OAG-BERT: Pre-train Heterogeneous Entity-augmented Academic Language Models

Xiao Liu[†*], Da Yin[†*], Xingjian Zhang[†], Kai Su[†], Kan Wu[†], Hongxia Yang[‡], Jie Tang[†**]
[†] Department of Computer Science and Technology, Tsinghua University, China
[‡] DAMO Academy, Alibaba Group, China
{liuxiao17,yd18}@mails.tsinghua.edu.cn,yang.yhx@alibaba-inc.com,jietang@tsinghua.edu.cn

ABSTRACT

To enrich language models with domain knowledge is crucial but difficult. Based on the world's largest public academic graph Open Academic Graph (OAG), we pre-train an academic language model, namely OAG-BERT, which integrates massive heterogeneous entities including paper, author, concept, venue, and affiliation. To better endow OAG-BERT with the ability to capture entity information, we develop novel pre-training strategies including heterogeneous entity type embedding, entity-aware 2D positional encoding, and span-aware entity masking. For zero-shot inference, we design a special decoding strategy to allow OAG-BERT to generate entity names from scratch. We evaluate the OAG-BERT on various downstream academic tasks, including NLP benchmarks, zero-shot entity inference, heterogeneous graph link prediction and author name disambiguation. Results demonstrate the effectiveness of the proposed pre-training approach to both comprehending academic texts and modeling knowledge from heterogeneous entities. OAG-BERT has been deployed to multiple real-world applications, such as reviewer recommendations and paper tagging in the AMiner system. OAG-BERT[1] is also available to the public through the CogDL package.

KEYWORDS

Pre-training, Language Modeling, Knowledge Representation, Heterogeneous Graph

1 INTRODUCTION

Pre-trained language models such as BERT [8], GPT [36] and XLNet [44] substantially promote the development of natural language processing. Besides pre-training for general purposes, more and more language models are targeting at specific domains, such as BioBERT [25] for biomedical field and SciBERT [2] for academic field, which establish new state-of-the-art on many domain-related benchmarks such as named entity recognition [9, 31], topic classification [4, 18] and so on.

However, most of these models are only pre-trained over domain corpus, but ignore to integrate domain entity knowledge, which is crucial for many entity-related downstream tasks. In the author name disambiguation task, the affiliation of a paper could contribute by indicating the field-of-study of an author. For example, authors from Max Planck Institute may focus more on science and engineering rather than humanity. We may also produce fine-grained fields-of-study out of the database, by leveraging information from title, abstract, affiliations, and even their author names. This will substantially boost the effectiveness for researchers to identify their related work of interest.

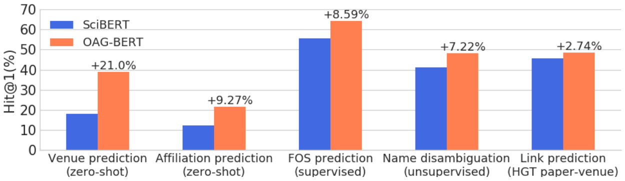

Figure 1: OAG-BERT outperforms SciBERT on a range of entity-related tasks by 2.75%-21.0% (Absolute Gain).

As a complement to corpora, many domain knowledge graphs are available to provide knowledge. For example, OAG [46] is the largest publicly available heterogeneous academic entity graph containing more than 700 million entities, including paper, author, field-of-study, venue and affiliation; and 2 billion relationships. If we can incorporate such tremendous amount of knowledge into domain language models, numerous downstream applications will be benefited. However, despite the abundant available knowledge sources, how to inject knowledge into language models has become the key problem. While many works have been concentrating on injecting the homogeneous entity and relation knowledge from large scale knowledge graphs [35, 39, 50], none of them considers the problem of heterogeneous scenario which usually holds true in practice.

To bridge the gap, we propose OAG-BERT to jointly model texts and heterogeneous entity knowledge via pre-training over the OAG. We collect around 5 million full papers and 110 million abstracts as corpora, and 70 million paper ego-networks including its authors, fields of study, venues and affiliations. To handle the heterogeneity, we design the heterogeneous entity type embeddings for each type of entity respectively. To implement the masked language pre-training over entity names with various lengths, we design a novel span-aware entity masking strategy that can select to mask a continuous span of tokens according to entity lengths. To better "notice" the OAG-BERT with the entity spans and sequence order, we propose the entity-aware 2D positional encoding to take both the inter-entity sequence order and intra-entity token order into consideration.

We first evaluate the OAG-BERT on various types of downstream applications, including traditional academic NLP datasets [2, 5], novel zero-shot entity inference [34], heterogeneous graph learning (link prediction) [10, 16] and author name disambiguation [3, 49]. For zero-shot inference, we develop a special decoding strategy

[1] https://github.com/thudm/oag-bert
[*] These authors contributed equally to this work.
[**] Jie Tang is the corresponding author.

for OAG-BERT, allowing it to generate fluent sequences like GPTs. Not only do the experiment results demonstrate OAG-BERT's competitive performance to previous pre-trained models on ordinary language benchmarks, but also support its outstanding grasp of entity knowledge by outperforming over tasks that heavily depend on entity knowledge.

To sum up, we make the following contributions in this paper:

- We propose to study the problem of enriching pre-trained language models with heterogeneous entity knowledge. To solve the problem, we design heterogeneous entity type embedding, span-aware entity masking and entity-aware 2D positional encoding. We also develop a special decoding strategy for BERT-style models to generate high-quality entities from scratch.
- We present the OAG-BERT, an entity knowledge augmented academic language model that is pre-trained over 5 million paper full-text, 110 million paper abstracts and billions of academic entities and relations from the OAG. It has the similar number of parameters with other BERT-based models such as SciBERT.
- We conduct relatively extensive experiments to demonstrate OAG-BERT's capability of traditional language tasks, zero-shot inference, heterogeneous graph learning and author name disambiguation. OAG-BERT has been deployed as the infrastructure of AMiner system[2] for OAG downstream applications. It is open to the public access through the CogDL [45] package.
- We apply the pre-trained OAG-BERT model to several real-world applications, such as the reviewer recommendation. It is also employed as a fundamental component in the AMiner system, which is further used to improve the performance on tasks like automatic paper tagging or author name disambiguation.
- We release the pre-trained OAG-BERT model in CogDL package for open access and free use.

2 RELATED WORKS

Our proposed OAG-BERT model is based on BERT [8], a self-supervised [28] bidirectional language model. It employs multi-layer transformers as its encoder and uses masked token prediction as its objective, which allows using massive unlabeled text data as training corpus. The model architecture and training scheme have been shown to be effective on various natural language tasks, such as question answering or natural language inference.

BERT has many variants. Some focus on the robustness of the pre-training process, like RoBERTa [29]. Some others try to incorporate more knowledge into the natural language pre-training. SpanBERT [17] develops span-level masking which benefits span selection tasks. ERNIE [50] introduces explicit knowledge graph inputs to the BERT encoder and achieves significant improvements over knowledge-driven tasks.

As for the academic domain, previous works such as BioBERT [25] or SciBERT [2] leverage the pre-training process on scientific domain corpus and achieve state-of-the-art performance on several academic NLP tasks. The S2ORC-BERT [30], applies the same method with SciBERT on a larger scientific corpus and slightly improves the performance on downstream tasks. Later works [14] further show that continuous training on specific domain corpus also benefits the downstream tasks. These academic pre-training models rely on large scientific corpora. SciBERT uses the semantic scholar corpus [1]. Other large academic corpora including AMiner [40], OAG [40, 46], and Microsoft Academic Graph (MAG) [19] also integrate massive publications with rich graph information as well, such as authors and research fields.

On academic graphs, there are some tasks that involve not only text information from papers but also structural knowledge lying behind graph links. For example, to disambiguate authors with the same names [3, 49], the model needs to learn node representations in the heterogeneous graph. To better recommend papers for online academic search [11, 12], graph information including related academic concepts and published venues could provide great benefits. To infer experts' trajectory across the world [43], associating authors with their affiliation on semantic level would help. Capturing features from paper titles or abstracts is far from enough for these types of challenges.

Targeting at graph-based problems, many graph representation learning methods were proposed in the last decade. Works like node2vec [13] and ProNE [47] focus on purely homogeneous graph structures and metapath2vec [10] later extends the idea to heterogeneous graphs. Neural-based methods like GCN [23] successfully introduce neural networks to solve the graph learning problem. Recent works including Heterogeneous Graph Transformer [16] and GPT-GNN [15] similarly borrow the idea from the natural language community, applying transformer blocks and pre-training scheme on graph tasks.

3 METHODS

The proposed OAG-BERT is a bidirectional transformer-based pre-training model. It can encode scientific texts and entity knowledge into high dimensional embeddings, which can be used for downstream tasks such as predicting the published venue for papers. We build the OAG-BERT model on top of the conventional BERT [8] model with 12 transformer [42] encoder layers.

While the original BERT model only focuses on natural language, our proposed OAG-BERT also incorporates heterogeneous entity knowledge. In other words, in addition to learning from pure scientific texts such as paper title or abstract, the OAG-BERT model can comprehend other types of information, such as the published venues or the affiliations of paper authors. To achieve that, we made several modifications to the model architecture and the pre-training process. We will introduce them in the following sections. An overview of the proposed OAG-BERT model is depicted in Figure 2.

3.1 Model Architecture

The key challenge for OAG-BERT lies in how to integrate knowledge into language models. Previous approaches [27, 50] mainly focus on injecting homogeneous entities and relations from knowledge graphs like Wikidata, and very few of them look into situations where there are heterogeneous entities.

To augment OAG-BERT with various types of entity knowledge, we place title, abstract along other entities from the same paper in a single sequence as one training instance (see Figure 2).

[2] https://www.aminer.cn

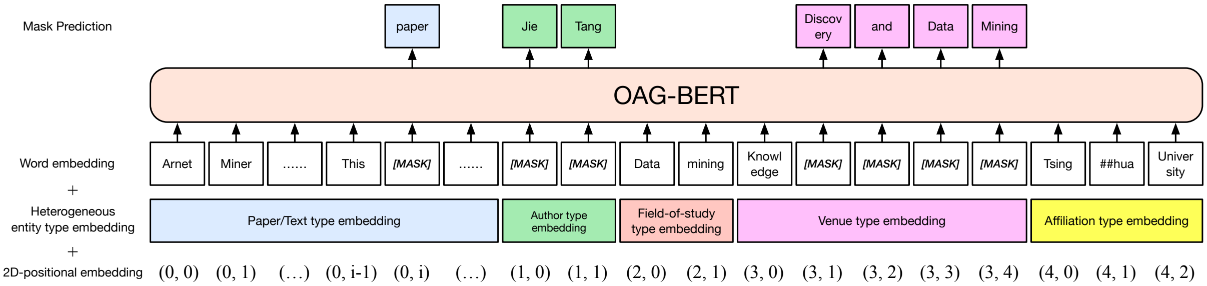

Figure 2: Heterogeneous entity augmentation in OAG-BERT. 1) For different entity types, we design heterogeneous entity type embedding. 2) For comparatively long entities (such as the "Knowledge Discovery and Data Mining"), we leverage the span-aware entity masking strategy, which selects a continuous span in the entity. 3) For positional embeddings across different entities, we design an entity-aware 2D-positional embedding strategy, whose first dimension is designed for indicating inter-entity sequence order, and the second dimension is designed for indicating intra-entity token sequence order.

There are five types of entities in total. We treat the text features (title and abstract) of a paper as one special text entity. The published venue, authors, affiliations, and research fields are the rest four types of entities. Following the notation in OAG, we use FOS (field-of-study) to denote research fields. Thanks to OAG, for venues, authors, affiliations, and FOS, their names have been cleaned up, deduplicated, and unified, which enables OAG-BERT to learn a consistent representation for each entity.

All the entities from one paper are concatenated as an input sample. To help the OAG-BERT model distinguish them, we use another three techniques: *Heterogeneous entity type embedding*, *Entity-aware 2D-positional encoding* and *Span-aware entity masking*.

Heterogeneous entity type embedding. The original BERT employs the next sentence prediction loss (NSP) to learn the relationship between sequences, which requires the use of token type embeddings to distinguish two sequences from each other. Tokens from two sequences are added by different token type embeddings. However, NSP loss is believed to harm rather than improve BERT's performance, as found in later works [17, 29]. Therefore, in this work, we abandon the NSP loss and discard the old token type embeddings.

On the other hand, in order to distinguish different types of entities, we propose to leverage entity type embedding in the pretraining process to indicate entity type, whose usage is similar to the token type embedding used in BERT.

For example, given the title and abstract of a paper "ArnetMiner: extraction and mining of academic social networks", we retrieve its authors, fields of studies, venues, and affiliation entities and concatenate them into a sequence less than 512 tokens. For pure text (such as title and abstract), we label them with the original entity type index (e.g., 0) to acquire its entity type embedding. For author entities (such as Jie Tang), we label them with author type index (e.g., 1). So are for other entities. What's more, because entities are order-invariant in the sequences, we shuffle their order in a sample sequence to avoid our model to learn any positional biases of these entities.

Entity-aware 2D-positional encoding. Although the transformer [42] architecture has achieved great success in sequence-based tasks, it is also known that the transformer itself is permutation-invariant (i.e. is not aware of the sequence order). The critical technique of applying transformer to natural language is to add a *positional embedding* to indicate the sequence order, including the absolute positional embedding used in vanilla Transformer [42] and BERT [8], and the relative positional embedding developed in Transformer-XL [6] and XLNet [44].

However, when we want OAG-BERT to capture entity knowledge, neither of them is applicable. This is because the conventional positional embedding can not distinguish words from entities that are adjacent to each other and of the same type. For instance, if there are two affiliations "Tsinghua University" and "Unviersity of California" being placed next to each other in a sequence, the transformer would assume that there is an affiliation named "Tsinghua University University of California".

To sum up, our requests could be summarized to two points: 1) the positional embedding should imply the *inter-entity* sequence order (which is used to distinguish different entities) and 2) the positional embedding should indicate the *intra-entity* token sequence order (which is used as the traditional positional embedding).

In light of this, we design the entity-aware 2D-positional embedding that solves both the inter-entity and intra-entity problem (see Figure 2). The first dimension is for inter-entity order, indicating the token is in which entity; the second dimension is for intra-entity order, indicating the sequence of tokens. For a given position, the final positional embedding is calculated by adding the two positional embeddings together.

Span-aware entity masking. When performing masking, for pure text contents such as paper title and abstract, we adopt the same random masking strategy as in BERT. However, for entities such as author names, field of study, venues and affiliations, to encourage OAG-BERT to memorize them, we develop a span-aware entity masking strategy which combines the advantages of both ERNIE [50] and SpanBERT [17].

The intuition of using this strategy is that, some of the entities are too long and thus too difficult for the OAG-BERT to learn. The span-aware entity masking strategy not only alleviates the problem, but also still preserves the sequential relationship of an entity's tokens: for entity that has less than 4 tokens, we will mask the whole entity; and for others, we sample masked lengths from a geometric distribution $\text{Geo}(p)$ which satisfies:

$$p = 0.2 \text{ , and } 4 \leq \text{Geo(p)} \leq 10 \tag{1}$$

If the sampled length is less than the entity length, we will only mask out the entity. For text contents and entity contents, we mask 15% of the tokens for each respectively.

Pre-LN BERT. Except for the previous changes to the original BERT architecture, we further adopt the Pre-LN BERT as used in deepspeed [37] , where layer normalization is placed inside the residual connection instead of after the add-operation in Transformer blocks. Previous work [48] demonstrates that training with Pre-LN BERT avoids vanishing gradients when using aggressive learning rates. Therefore, it is shown to be more stable than the traditional Post-LN version for optimization.

3.2 Pre-training Details

The pre-training of OAG-BERT is separated into two stages. In the first stage, we only use scientific texts (paper title, abstract, and body) as the model inputs, without using the entity augmented inputs introduced above. This process is similar to the pre-training of the original BERT model. We name the intermediate pre-trained model as the vanilla version of OAG-BERT. In the second stage, based on the vanilla OAG-BERT, we continue to train the model on the heterogeneous entities, including title, abstract, venue, authors, affiliations, and field-of-studies (FOS).

First Stage: Pre-train the vanilla OAG-BERT. In the first stage of pre-training, we construct the training corpus from two sources: one comes from the PDF storage of AMiner, which mainly consists of arXiv PDF dumps; the other comes from the PubMed XML dump. We clean up and sentencize the corpus with SciSpacy [32]. The corpus adds up to around 5 million unique paper full-text from multiple disciplines. In terms of vocabulary, we construct our OAG-BERT vocabulary using WordPiece, which is also used in original BERT implementation. This ends up with 44,000 unique tokens in our vocabulary.

For better handling the entity knowledge of authors in the OAG, in the data prepossessing we transform the author name list as a sentence for each paper and place it between the title and abstract. Therefore, compared to previous models like SciBERT, our vocabulary contains more tokens from author names.

Following the training procedures of BERT, the vanilla OAG-BERT is first pre-trained on samples with a maximum of 128 tokens. After the loss has converged, we shift to pre-training it over samples with 512 tokens.

Second Stage: Enrich OAG-BERT with entity knowledge. In the second stage of pre-training, we use papers and related entities from the OAG corpus. Compared to the corpus used in the first stage, we do not have full texts for all papers in OAG. Thus, we only use paper title and abstract as the paper text information. From this corpus, we picked all authors with at least 3 papers published. Then we filtered out all papers not linked to these selected authors. Finally, we got 120 million papers, 10 million authors, 670 thousand FOS, 53 thousand venues, and 26 thousand affiliations. Each paper and its connected entities are concatenated into a single training instance, following the input construction method described above. In this stage, we integrate the three strategies mentioned in Section 3.1 to endow OAG-BERT the ability to "notice" the entities, rather than regarding them as pure texts.

Our pre-training is conducted with 32 Nvidia Tesla V100 GPUs and an accumulated batch size of 32768. We use the default BERT pre-training configurations in deepspeed. We run 16K steps for the first stage pre-training and another 4K steps for the second stage.

4 EXPERIMENTS

In this section, we will introduce several experiments to demonstrate the effectiveness of our proposed OAG-BERT. First, to exhibit how OAG-BERT works on multi-type information, we design intuitive zero-shot inference tasks. Then, we make extensions to supervised classification tasks. We further apply the pre-trained embeddings to name disambiguation and link prediction tasks, which present the superior capability of OAG-BERT in leveraging various types of entities. Finally, on the NLP tasks used by SciBERT, we additionally verify that the proposed OAG-BERT model can also achieve competitive results with text-only information provided. An overview of the model performance is shown in Table 1.

4.1 Zero-shot Inference

Although not using unidirectional decoder structure like GPT-3, we find that the bidirectional encoder-based OAG-BERT is also capable of decoding entities based on the knowledge it learned during the pre-training process. We develop a simple extension to the Masked Language Model (MLM) to achieve that.

In MLM, the token prediction task in the pre-training process can be seen as maximizing the probability of masked input tokens. It treats the predictions for each token as independent processes. The target can be denoted as maximizing $\sum_{w \in masked} \log P(w|C)$, where *masked* is the collection of masked tokens and C denotes contexts, which represents the inputs of MLM, including both input tokens and position information.

In the entity decoding process, we cannot ignore the dependencies between tokens in each entity, which requires us to jointly consider the probability of all tokens in one entity as following $\log P(w_1, w_2, ..., w_l|C)$, where l is the entity length and w_i is the i-th token in the entity. As MLM is not unidirectional model, the decoding order for the tokens in one entity can be arbitrary. Suppose the decoding order is $w_{i_1}, w_{i_2}, ..., w_{i_l}$, where $i_1, i_2, ..., i_l$ is a permutation of $1, 2, ..., l$. Then the prediction target can be reformed as maximizing

$$\sum_{1 \leq k \leq l} \log P(w_{i_k}|C, w_{i_1}, w_{i_2}, ..., w_{i_{k-1}}) \tag{2}$$

However, the number of possible decoding orders is $l!$, which makes it extremely expensive to calculate while dealing with long entities. Thus, we adopt two strategies to solve this problem. First, while calculating the probability for one given entity, we use greedy selection to decide the decoding order. In other words, for each round of decoding, we choose the token with maximal probability to decode. An example is depicted in Figure 3. Second, when decoding an entity from scratch, we use beam search [41] to search the token combinations with the highest probability.

Another challenge for decoding using the MLM model is to choose the appropriate entity length. Instead of using fixed length

Table 1: The summary of model performance for all tasks. We report the performance of only using paper titles as inputs in *title-only* and the best performance of using other features such as FOS or venue as inputs in *mixed*.

Method		Zero-shot Inference[1]			Supervised Classification[2]			NA[3]	Link Prediction[4]	
		FOS	Venue	Affiliation	FOS	Venue	Affiliation		Paper-Field	Paper-Venue
SciBERT	*title-only*	29.59%	10.03%	8.00%	55.13%	61.86%	35.44%	0.3690	0.4740	0.4570
	mixed	35.33%	18.00%	12.40%	55.63%	78.05%	56.04%	0.4101	-	-
OAG-BERT	*title-only*	37.33%	22.67%	11.77%	54.54%	63.03%	35.04%	0.4120	**0.4892**	**0.4844**
	mixed	**49.59%**	**39.00%**	**21.67%**	**64.22%**	**78.47%**	**57.63%**	**0.4823**	-	-

[1,2] Hit@1 is reported for zero-shot inference and supervised classification.

[3] NA is short for Name disambiguation. The macro pairwise f1 score is reported.

[4] For link prediction tasks, we use pre-trained models to encode all types of nodes. Only title was provided for paper nodes. NDCG is reported.

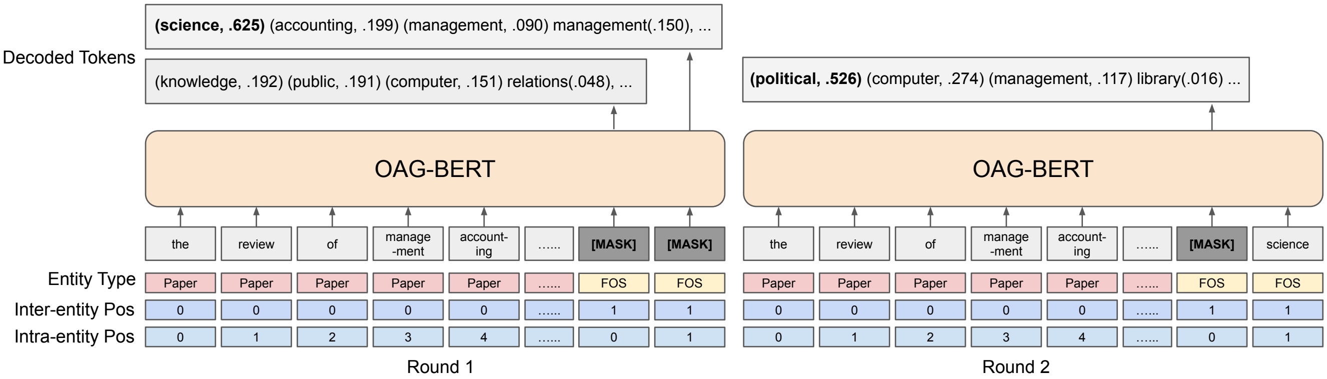
Decoded Tokens
(science, .625) (accounting, .199) (management, .090) management(.150), ...
(knowledge, .192) (public, .191) (computer, .151) relations(.048), ...
(political, .526) (computer, .274) (management, .117) library(.016) ...
OAG-BERT
OAG-BERT
the
review
of
manage -ment
account- ing
......
[MASK]
[MASK]
science
Entity Type
Paper
FOS
Inter-entity Pos
Intra-entity Pos
0
1
2
3
4
Round 1
Round 2

Figure 3: The decoding process of OAG-BERT. The left figure indicates that OAG-BERT decodes the masked token "science" at the second position with the highest probability (0.625) for the first round. Then it decodes "political" at the first position with highest probability (0.526) for the second round as shown in the right figure.

while decoding from scratch, we traverse all entity lengths in a pre-defined range depending on the entity type and choose top candidates according to the calculated probability in Equation 2.

We design three zero-shot inference tasks to evaluate the entity generation capability of our proposed OAG-BERT and make comparisons with SciBERT, the current state-of-the-art pre-training model in academic domain.

Field-of-Study (FOS) Inference To evaluate the performance of decoding field-of-study, we adopt the research field prediction task from MAG (Microsoft Academic Graph) [38]. First, we choose 19 top-level field-of-studies (FOS) such as "biology" and "computer science". Then, from the paper data which were not used in the pre-training process, we randomly select 1,000 papers for each FOS. The task is to predict which research field each paper belongs to.

For each paper, we estimate the probabilities for all FOS candidates and choose the top one. When estimating each one, we concatenate the FOS candidate with the paper title as model input and mask the FOS candidate. For example, when estimating the probability of "computer science", we add two "[MASK]" tokens to the end of the original title as the input. For OAG-BERT, we treat the newly added "[MASK]" tokens as a new entity, reset entity positions and use FOS entity type embedding additionally. Then we use Equation 2 to calculate the probability. This is denoted as the *Plain* method, as depicted in Figure 3.

We also apply two techniques to improve the model decoding performance. The first technique is to add extra *prompt* word to the end of the paper title (before masked tokens). We select "Field of study:" as the prompt words in the FOS inference task. The second technique is to concatenate the paper abstract to the end of the paper title.

Venue and Affiliation Inference Similar to the FOS inference task, we create venue and affiliation inference tasks. From non-pretrained papers, we choose 30 most frequent arXiv categories and 30 affiliations as inference candidates, with 100 papers randomly selected for each candidate. Full lists of the candidates including FOS candidates are enclosed in the appendix.

The experiment settings completely follow the FOS inference task, except that we use "Journal or Venue:" and "Affiliations:" as prompt words respectively. The entity type embeddings for masked entities in OAG-BERT are also replaced by venue and affiliation entity type embeddings accordingly. We report the Hit@1 and MRR scores in Table 2.

Results Analysis In Table 2, we can see that the proposed augmented OAG-BERT outperforms SciBERT by a large margin. Although SciBERT was not pre-trained with entity knowledge, it still performs much greater than a random guess, which means the inference tasks are not independent of the paper content information. We speculate that the pre-training process on paper content (as used in SciBERT) also helps the model learn some generalized knowledge on other types of information, such as field-of-studies or venue names.

Table 2: The results for zero-shot inference tasks.

Method	FOS		Venue		Affiliation	
	Hit@1	MRR	Hit@1	MRR	Hit@1	MRR
SciBERT	19.93%	0.37	9.87%	0.22	6.93%	0.19
+prompt	29.59%	0.47	10.03%	0.21	8.00%	0.20
+abstract	25.66%	0.43	18.00%	0.32	10.33%	0.22
+both	35.33%	0.52	9.83%	0.22	12.40%	0.25
OAG-BERT	34.36%	0.51	21.00%	0.37	11.03%	0.24
+prompt	37.33%	0.55	22.67%	0.39	11.77%	0.25
+abstract	**49.59%**	**0.67**	**39.00%**	**0.57**	**21.67%**	**0.38**
+both	49.51%	**0.67**	38.47%	**0.57**	21.53%	**0.38**

We also observe that the proposed use of abstract can always help improve the performance. On the other hand, the prompt words works well with SciBERT but only provide limited help for OAG-BERT. Besides, the affiliation inference task appears to be harder than the other two tasks. Further analysis are provided in the A.1. Two extended experiments are enclosed as well, which reveal two findings:

(1) Using the summation of token log probabilities as the entity log probability is better than using the average.
(2) The out-of-order decoding is more suitable for encoder-based models like SciBERT and OAG-BERT, as compared with the left-to-right decoding.

Table 3: The generated FOS for the paper of GPT-3. The gold FOS are bolded. FOS not in the original OAG FOS candidate list are underlined.

Title	Language Models are Few-Shot Learners
Abstract	Recent work has demonstrated substantial gains on many NLP tasks and benchmarks by pre-training on a large corpus of text followed by fine-tuning on a specific task. While typically task-agnostic in architecture, this method still requires task-specific fine-tuning datasets of thousands or tens of thousands of examples. By contrast, humans can generally...
Generated FOS	Natural language processing, Autoregressive language model, **Computer science**, Sentence, Artificial intelligence, Domain adaptation, **Language model**, Few shot learning, Large corpus, Arithmetic, Machine learning, Architecture, Theoretical computer science, Data mining, **Linguistics**, Artificial language processing
Gold FOS	Language model, Computer science, Linguistics

Case Study To exhibit the capability of decoding entities, we applied the method described above on the task of FOS generation. Given the paper title and abstract, we use beam search with a width of 16 to decode FOS entities. We search from single-token entities to quadruple-token entities. The top 16 generated ones are listed in Table 3. In Table 3, the gold FOS are all in the top 16. Some fine-grained entities, though not in candidates, are also generated, such as *Autoregressive language model* or *Few shot learning*.

However, we can still observe some ill-formed or inappropriate entities such as *Architecture* or *Artificial language processing*. While the paper is related to *Model architecture*, the single-token *Architecture* usually refers to the science of designing buildings, which is not suitable in this case. *Artificial language model*, on the other hand, is more like a combination of *Artificial Intelligence* and *Language model*, which have already been generated.

Table 4: The results of the classification task.

Tasks	Freeze		Finetune	
	SciBERT	OAG-BERT	SciBERT	OAG-BERT
FOS				
title only	33.25$^{0.25}$	**43.28**$^{0.12}$	**55.13**$^{0.30}$	54.54$^{0.29}$
+author	30.15$^{0.07}$	**41.87**$^{0.06}$	**55.63**$^{0.42}$	55.30$^{0.43}$
+venue	34.77$^{0.17}$	**46.99**$^{0.15}$	63.18$^{0.18}$	**63.53**$^{0.08}$
+aff	32.83$^{0.13}$	**43.07**$^{0.11}$	**55.06**$^{0.21}$	54.65$^{0.38}$
+all	32.83$^{0.08}$	**45.47**$^{0.16}$	63.43$^{0.15}$	**64.22**$^{0.38}$
Venue				
title only	24.62$^{0.52}$	**32.87**$^{1.47}$	61.86$^{0.32}$	**63.03**$^{0.46}$
+author	21.21$^{0.82}$	**30.91**$^{0.96}$	62.62$^{0.34}$	**63.46**$^{0.48}$
+aff	24.38$^{0.49}$	**32.32**$^{1.36}$	62.13$^{0.43}$	**62.65**$^{0.49}$
+fos	40.49$^{1.25}$	**52.61**$^{0.79}$	78.05$^{0.14}$	**78.47**$^{0.25}$
+all	39.92$^{1.17}$	**51.33**$^{0.44}$	77.88$^{0.16}$	**78.34**$^{0.62}$
Affiliation				
title only	13.88$^{0.83}$	**19.72**$^{0.64}$	**35.44**$^{0.45}$	35.04$^{0.61}$
+author	20.65$^{1.04}$	**32.19**$^{0.92}$	52.68$^{0.18}$	**53.33**$^{0.43}$
+venue	16.57$^{0.60}$	**25.23**$^{0.72}$	43.13$^{0.36}$	**43.65**$^{0.40}$
+fos	17.39$^{0.86}$	**22.06**$^{0.37}$	37.05$^{0.80}$	**37.60**$^{0.51}$
+all	24.02$^{0.87}$	**32.49**$^{0.50}$	56.04$^{0.95}$	**57.63**$^{0.49}$

In summary, although our proposed OAG-BERT model is not born for decoding, it still exhibits the potential of generating high-quality entities in the zero-shot settings.

4.2 Supervised Classification

In this section, we develop the supervised classification tasks on top of the datasets described above, which are enlarged by 10 times following the same generating process. The data in the zero-shot inference are kept as test sets. We construct validation sets to select the best models during fine-tuning, with the same size as the test sets. The rest data are used as training sets. The sizes of all datasets for all tasks are enclosed in the appendix.

In supervised classification tasks, we remove the masked tokens and feed the averaged output embeddings from the pre-training models to a single fully-connected layer. We apply softmax layer to make predictions at last. As for the inputs, to present the effectiveness of heterogeneous entity types, we not only use paper titles as inputs but also concatenate other entities. Besides, we also tested the model performance with and without the original pre-training model parameters frozen. We follow the standard configurations for fine-tuning BERT, which are enclosed in the appendix.

As shown in Table 4, the OAG-BERT outperforms SciBERT by a large margin when the parameters in pre-trained parts are frozen. When not frozen, for venue and affiliation prediction, OAG-BERT surpasses SciBERT significantly. In FOS prediction, although OAG-BERT under-performs SciBERT in some cases, the best performance for using all available entities in OAG-BERT still beats the one reached by SciBERT.

We also observe that different types of entities contribute to various tasks in dissimilar ways. For example, the use of author information is particularly helpful for affiliation prediction but not very useful in FOS prediction. On the other hand, the field of study (FOS) inputs, work pretty well in venue prediction but provide marginal improvements to affiliation prediction.

In conclusion, the proposed OAG-BERT is effective in both zero-shot tasks and supervised tasks. The additionally learned heterogeneous entities can help the model reach better performance while dealing with multiple types of inputs.

Table 5: The Macro Pairewise F1 scores for the name disambiguation task.

Inputs	SciBERT	OAG-BERT
title	0.3690	**0.4120**
+fos	0.4101	**0.4643**
+venue	0.3603	**0.4247**
+fos+venue	0.3903	**0.4823**
Leader Board Top 1		0.4900

4.3 Name Disambiguation

Previous experiments focus on the decoding capability and fine-tuning performance on downstream tasks. In this section, we adopt the name disambiguation problem to validate the paper representation quality produced by OAG-BERT. In this problem, given a set of papers with authors of the same name, the designed algorithm needs to separate these papers into several clusters, where papers in the same cluster belong to the same author and different clusters represent different authors.

We use the public dataset *whoiswho-v1*[3] [3, 49] and apply the embeddings generated by pre-trained models to solve name disambiguation from scratch. Formally, for each paper, we use the paper title and other attributes such as field-of-study or published venue as model input. Then we average over all the output token embeddings for the paper and use it as the paper embedding. After that, we build a graph with all papers as the graph nodes and set a threshold to select edges. The edges are between papers where the pairwise cosine similarity of their embeddings is larger than the threshold. Finally, for each connected component in the graph, we treat it as a cluster. We searched the thresholds from 0.65 to 0.95 on the validation set. The threshold from the best validation results is used on test set evaluation. We calculated the macro pairwise f1 score following previous works.

The results in Table 5 indicate that the embedding of OAG-BERT is significantly better than the SciBERT embedding while directly used in the author name disambiguation. We also observe that for SciBERT the best threshold is always 0.8 while this value for OAG-BERT is 0.9, which reflects that the paper embeddings produced by OAG-BERT are generally closer than the ones produced by SciBERT.

In Table 5 we only list the results with title, field-of-study, and venue as inputs. Though we attempted to use the abstract, author, and affiliation information, there is no performance improvement as expected. We speculate it is because these types of information are more complex to use, which might require additional classifier head or fine-tuning, as the supervised classification task mentioned above. In addition, we also report the top 1 score in the name disambiguation challenge leaderboard[4] and find that our proposed OAG-BERT reaches close performance as compared with the top 1.

4.4 Link Prediction

In previous sections, we present the effectiveness of using OAG-BERT individually. In this section, we apply the heterogeneous entity embeddings of OAG-BERT as pre-trained initializations for node embeddings on the academic graph and show that OAG-BERT can also work together with other types of models. Specifically, we take the heterogeneous graph transformer (HGT) model from [16] and combine it with the pre-trained embeddings from OAG-BERT.

To make predictions for the links in the heterogeneous graph, the authors of HGT first extract node features and then apply HGT layers to encode graph features. For paper nodes, the authors use XLNet [44] to encode titles as input features. For other types of nodes, HGT use metapath2vec [10] to initialize the features.

However, there are two problems with using XLNet on the heterogeneous academic graph. First, the XLNet was pre-trained on universal language corpus, which is lack of academic domain data. Second, XLNet can only encode paper nodes by using their titles and is unable to generate useful embeddings for other types of nodes like author or affiliation.

To this end, we propose to replace the original XLNet encoder with our OAG-BERT model, which can tackle the two challenges mentioned above. We use the OAG-BERT model to encode all types of nodes and use the generated embeddings as their node features. To prove the effectiveness of OAG-BERT on encoding heterogeneous nodes, we also compare the performance of SciBERT with OAG-BERT. We experimented on the CS dataset released by HGT[5]. The details of the dataset are delivered in the appendix.

The NDCG and MRR scores for the Paper-Field and Paper-Venue link prediction are reported in Table 7. It shows that SciBERT surpasses the original XLNet performance significantly, due to the pre-training on the large scientific corpus. Our proposed OAG-BERT made further improvements on top of that, as it can better understand the entity knowledge on the heterogeneous graph.

4.5 NLP Tasks

Previous experiments have demonstrated the superiority of OAG-BERT on tasks involving multi-type entities. In this section, we will further explore the performance of OAG-BERT on natural language processing tasks, which only contain text-based information such as paper titles and abstracts. We will show that although pre-trained with heterogeneous entities, the OAG-BERT can still perform competitive results with SciBERT on NLP tasks.

We made comparisons over three models, including **SciBERT** (both the original paper results and the reproduced results), **S2ORC** (similar to SciBERT except pre-trained with more data), and **OAG-BERT** (both the vanilla version and the augmented version).

In accord with SciBERT [2], we evaluate the model performance on the same 12 NLP tasks, including Named Entity Recognition (NER), Dependency Parsing (DEP), Relation Extraction (REL), PICO Extraction (PICO), and Text Classification (CLS). These tasks only focus on single sentence representation so we add another three sequential sentence classification (SSC) tasks used in [5], to further verify the capability of pre-training models on long texts. The evaluation metrics are also accord with the usage in SciBERT [2] and

[3] https://www.aminer.cn/whoiswho

[4] https://www.biendata.xyz/competition/aminer2019/leaderboard/

[5] https://github.com/acbull/pyHGT

Table 6: The results for NLP Tasks.

Field	Task	Dataset	Samples[1]	S2ORC	SciBERT		OAG-BERT	
					Reported[2]	Reproduced	Vanilla	Augmented
Bio	NER	BC5CDR [26]	3942	$90.04^{0.06}$	90.01	$89.77^{.23}$	$89.71^{.13}$	$89.71^{.12}$
		JNLPBA [21]	16807	$77.70^{.25}$	77.28	$77.29^{.38}$	$75.81^{.20}$	$76.99^{.04}$
		NCBI-disease [9]	5424	$88.70^{.52}$	88.57	$88.10^{.06}$	$87.90^{.12}$	$88.77^{.56}$
	PICO	EBM-NLP [33]	27879	$72.35^{.95}$	72.28	$72.52^{.71}$	$72.22^{.24}$	$71.74^{.49}$
	DEP	GENIA - LAS [20]	14326	$90.80^{.19}$	90.43	$90.57^{.08}$	$89.99^{.10}$	$90.12^{.11}$
		GENIA - UAS [20]		$92.31^{.18}$	91.99	$92.12^{.07}$	$91.57^{.08}$	$91.63^{.09}$
	REL	ChemProt [24]	4169	$84.59^{.93}$	83.64	$83.46^{.28}$	$82.14^{1.12}$	$80.21^{1.42}$
	SSC	Pubmed-RCT-20k [7]	15130	-	92.90	$92.86^{.12}$	$92.80^{.05}$	$92.73^{.08}$
		NICTA-piboso [22]	735	-	84.80	$83.93^{.58}$	$83.02^{.67}$	$84.00^{.32}$
CS	NER	SciERC [31]	1861	$68.93^{.19}$	67.57	$66.28^{.20}$	$67.80^{.24}$	$66.75^{.77}$
	REL	SciERC [31]	3219	$81.77^{1.64}$	79.97	$80.21^{.88}$	$76.59^{0.75}$	$78.63^{.06}$
	CLS	ACL-ARC [18]	1688	$68.45^{2.47}$	70.98	$70.34^{3.07}$	$66.13^{1.58}$	$64.79^{3.35}$
	SSC	CSAbstruct [5]	1668	-	83.10	$82.40^{.33}$	$82.48^{.44}$	$82.59^{.67}$
Multi	CLS	Paper Field [38]	84000	$65.99^{.08}$	65.71	$65.77^{.13}$	$64.67^{.14}$	$64.95^{.10}$
		SciCite [4]	7320	$84.76^{.37}$	85.49	$85.65^{.54}$	$85.25^{.38}$	$84.95^{.32}$

[1] *Samples* refers to the number of training samples in the dataset.

[2] We run the fine-tuning process for 5 times with different random seeds and report the mean and standard deviation. The results in the original paper of SciBERT do not report this. The results for NER tasks in the original SciBERT model use a different casing version of pre-trained model while all other results are achieved by uncased pre-trained models.

Table 7: The result of link prediction tasks.

Tasks	Paper-Field		Paper-Venue	
	NDCG	MRR	NDCG	MRR
XLNet	0.3939	0.4473	0.4385	0.2584
SciBERT	0.4740	0.5743	0.4570	0.2834
OAG-BERT	**0.4892**	**0.6099**	**0.4844**	**0.3131**

Sequential-Sentence-Classification [5], which can be found in the appendix along with the task details and hyper-parameter settings.

The results in Table 6 show that the proposed OAG-BERT is competitive with SciBERT and a bit behind the S2ORC. Comparing with the reproduced SciBERT, our vanilla OAG-BERT only shows clear disadvantages on the SciERC REL task and the ACL-ARC CLS task, where datasets are relatively small and are sensitive to a few swinging samples. We ascribe the minor differences in other tasks to the differences in training corpus and the data cleaning techniques. The augmented OAG-BERT, although trained with heterogeneous entities that differ from the inputs of downstream NLP tasks, still presents similar performance to the vanilla version.

In summary, despite the fact that the OAG-BERT does not surpass the previous state-of-the-art academic pre-training model on NLP tasks, it still keeps the knowledge on these language dedicated tasks even after pre-training with multiple types of entities.

5 DEPLOYED APPLICATIONS

In this section, we will introduce several real-world applications where our OAG-BERT is employed.

First, the results on the name disambiguation tasks indicate that the OAG-BERT is relatively strong at encoding paper information with multi-type entities, which further help produce representative embeddings for the paper authors. Thus, we apply the OAG-BERT to the reviewer recommendation problem.

To tackle this problem, we collaborate with Alibaba and develop a practical algorithm on top of the OAG-BERT which can automatically assign proper reviewers to applications and greatly benefits the reviewing process.

In addition to that, we also integrate the OAG-BERT as a fundamental component for the AMiner [40] system. In AMiner, we utilize OAG-BERT to handle rich information on the academic heterogeneous graph. For example, with the ability of decoding FOS entities, we use the OAG-BERT to automatically generate FOS candidates for unlabeled papers. Besides, we similarly amalgamate the OAG-BERT into the name disambiguation framework. Finally, we employ OAG-BERT to recommend related papers for users, leveraging its capability in encoding paper embeddings.

Moreover, we release the OAG-BERT model in CogDL package, helping users take advantages of our OAG-BERT model in their own applications.

6 CONCLUSTION

In conclusion, we propose a new pre-training model in the academic domain, called OAG-BERT. Compared to previous models like SciBERT, the OAG-BERT incorporates entity knowledge during pre-training, which benefits lots of downstream tasks that involve multi-type entities, such as name disambiguation or link prediction on the heterogeneous academic graph. We apply OAG-BERT to real-world applications, which improves the efficiency of these applications. We finally release the pre-trained model in CogDL, providing free use to arbitrary users.

There are still some problems remained. First, although OAG-BERT can decode entities, it is hard to generate long entities efficiently, due to the exhaustive search for the entity length. Second, the learning for sparse entities such as author names is much less effective than other entities due to the lack of pre-training, which hinders the downstream tasks to fully leverage the entity information. We leave these problems for future explorations.

REFERENCES

[1] Waleed Ammar, Dirk Groeneveld, Chandra Bhagavatula, et al. 2018. Construction of the literature graph in semantic scholar. *arXiv preprint arXiv:1805.02262* (2018).

[2] Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676* (2019).

[3] Bo Chen, Jing Zhang, Jie Tang, Lingfan Cai, Zhaoyu Wang, Shu Zhao, Hong Chen, and Cuiping Li. 2020. CONNA: Addressing Name Disambiguation on The Fly. *TKDE* (2020).

[4] Arman Cohan, Waleed Ammar, Madeleine Van Zuylen, and Field Cady. 2019. Structural scaffolds for citation intent classification in scientific publications. *arXiv preprint arXiv:1904.01608* (2019).

[5] Arman Cohan, Iz Beltagy, Daniel King, Bhavana Dalvi, and Daniel S Weld. 2019. Pretrained language models for sequential sentence classification. *arXiv preprint arXiv:1909.04054* (2019).

[6] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860* (2019).

[7] Franck Dernoncourt and Ji Young Lee. 2017. Pubmed 200k rct: a dataset for sequential sentence classification in medical abstracts. *arXiv preprint arXiv:1710.06071* (2017).

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[9] Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu. 2014. NCBI disease corpus: a resource for disease name recognition and concept normalization. *Journal of biomedical informatics* 47 (2014).

[10] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *SIGKDD*.

[11] Zhengxiao Du, Jie Tang, and Yuhui Ding. 2018. Polar: Attention-based cnn for one-shot personalized article recommendation. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer.

[12] Zhengxiao Du, Jie Tang, and Yuhui Ding. 2019. POLAR++: Active One-shot Personalized Article Recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2019).

[13] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *SIGKDD*.

[14] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. *arXiv preprint arXiv:2004.10964* (2020).

[15] Ziniu Hu, Yuxiao Dong, Kuansan Wang, Kai-Wei Chang, and Yizhou Sun. 2020. Gpt-gnn: Generative pre-training of graph neural networks. In *SIGKDD*.

[16] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. 2020. Heterogeneous graph transformer. In *WWW*.

[17] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *TACL* 8 (2020).

[18] David Jurgens, Srijan Kumar, Raine Hoover, Dan McFarland, and Dan Jurafsky. 2018. Measuring the evolution of a scientific field through citation frames. *TACL* 6 (2018).

[19] Anshul Kanakia, Zhihong Shen, Darrin Eide, and Kuansan Wang. 2019. A scalable hybrid research paper recommender system for microsoft academic. In *WWW*.

[20] J-D Kim, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. 2003. GENIA corpus - a semantically annotated corpus for bio-textmining. *Bioinformatics* 19, suppl_1 (2003).

[21] Jin-Dong Kim, Tomoko Ohta, Yoshimasa Tsuruoka, Yuka Tateisi, and Nigel Collier. 2004. Introduction to the bio-entity recognition task at JNLPBA. In *JNLPBA*. Citeseer.

[22] Su Nam Kim, David Martinez, Lawrence Cavedon, and Lars Yencken. 2011. Automatic classification of sentences to support evidence based medicine. In *BMC bioinformatics*, Vol. 12. Springer.

[23] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[24] Jens Kringelum, Sonny Kim Kjaerulff, Søren Brunak, Ole Lund, Tudor I Oprea, and Olivier Taboureau. 2016. ChemProt-3.0: a global chemical biology diseases mapping. *Database* 2016 (2016).

[25] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* 36, 4 (2020).

[26] Jiao Li, Yueping Sun, Robin J Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J Mattingly, Thomas C Wiegers, and Zhiyong Lu. 2016. BioCreative V CDR task corpus: a resource for chemical disease relation extraction. *Database* 2016 (2016).

[27] Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020. K-bert: Enabling language representation with knowledge graph. In *AAAI*, Vol. 34.

[28] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Zhaoyu Wang, Li Mian, Jing Zhang, and Jie Tang. 2020. Self-supervised learning: Generative or contrastive. *arXiv preprint arXiv:2006.08218* 1, 2 (2020).

[29] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[30] Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Dan S Weld. 2019. S2orc: The semantic scholar open research corpus. *arXiv preprint arXiv:1911.02782* (2019).

[31] Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. *arXiv preprint arXiv:1808.09602* (2018).

[32] Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. 2019. Scispacy: Fast and robust models for biomedical natural language processing. *arXiv preprint arXiv:1902.07669* (2019).

[33] Benjamin Nye, Junyi Jessy Li, Roma Patel, Yinfei Yang, Iain J Marshall, Ani Nenkova, and Byron C Wallace. 2018. A corpus with multi-level annotations of patients, interventions and outcomes to support language processing for medical literature. In *ACL*, Vol. 2018. NIH Public Access.

[34] Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066* (2019).

[35] Nina Poerner, Ulli Waltinger, and Hinrich Schütze. 2019. E-bert: Efficient-yet-effective entity embeddings for bert. *arXiv preprint arXiv:1911.03681* (2019).

[36] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019).

[37] Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *SIGKDD*.

[38] Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-June Hsu, and Kuansan Wang. 2015. An overview of microsoft academic service (mas) and applications. In *WWW*.

[39] Tianxiang Sun, Yunfan Shao, Xipeng Qiu, Qipeng Guo, Yaru Hu, Xuanjing Huang, and Zheng Zhang. 2020. CoLAKE: Contextualized Language and Knowledge Embedding. *arXiv preprint arXiv:2010.00309* (2020).

[40] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. 2008. Arnetminer: extraction and mining of academic social networks. In *SIGKDD*.

[41] C. Tillmann and H. Ney. 2003. Word Reordering and a Dynamic Programming Beam Search Algorithm for Statistical Machine Translation. *Computational Linguistics* 29 (2003), 97–133.

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762* (2017).

[43] Kan Wu, Jie Tang, and Chenhui Zhang. 2018. Where Have You Been? Inferring Career Trajectory from Academic Social Network.. In *IJCAI*.

[44] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237* (2019).

[45] Yan Wang Qibin Chen Yizhen Luo Xingcheng Yao Aohan Zeng Shiguang Guo Peng Zhang Guohao Dai Yu Wang Chang Zhou Hongxia Yang Jie Tang Yukuo Cen, Zhenyu Hou. 2021. CogDL: An Extensive Toolkit for Deep Learning on Graphs. *arXiv preprint arXiv:2103.00959* (2021).

[46] Fanjin Zhang, Xiao Liu, Jie Tang, Yuxiao Dong, Peiran Yao, Jie Zhang, Xiaotao Gu, Yan Wang, Bin Shao, Rui Li, et al. 2019. Oag: Toward linking large-scale heterogeneous entity graphs. In *SIGKDD*.

[47] Jie Zhang, Yuxiao Dong, Yan Wang, Jie Tang, and Ming Ding. 2019. ProNE: Fast and Scalable Network Representation Learning.. In *IJCAI*, Vol. 19.

[48] Minjia Zhang and Yuxiong He. 2020. Accelerating Training of Transformer-Based Language Models with Progressive Layer Dropping. *arXiv preprint arXiv:2010.13369* (2020).

[49] Yutao Zhang, Fanjin Zhang, Peiran Yao, and Jie Tang. 2018. Name Disambiguation in AMiner: Clustering, Maintenance, and Human in the Loop.. In *SIGKDD*.

[50] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. *arXiv preprint arXiv:1905.07129* (2019).

A EXPERIMENT SUPPLEMENTARY

A.1 Zero-shot Inference

Use of Prompt Word As shown in Table 2, the use of proposed prompt words in the FOS inference task, turns out to be fairly useful for SciBERT to decode paper fields (FOS). We conjecture it is because the extra appended prompt words can help alter the focus of the pre-training model while making predictions on masked tokens. However, the improvement for SciBERT is marginal on affiliation inference. When decoding venue, it even hurts the performance. This is probably due to the improper choice of prompt words.

For OAG-BERT, this technique has limited help as our expectation. Instead of using continuous positions as SciBERT, OAG-BERT encodes inter-entity positions to distinguish different entities and paper texts. Thus the additional appended prompt word is treated as part of the paper title and is not adjacent to the masked entities for OAG-BERT.

Use of Abstract The use of abstract can greatly improve the model inference performance in both SciBERT and OAG-BERT. Both models frequently accept long text inputs in the pre-training process, which makes them naturally favor abstracts. Besides, abstracts contain rich text information which can help the pre-training model capture the main idea of the whole paper.

Task Comparisons The affiliation generation task appears to be much harder than the other two tasks. This is probably due to the weak semantic information contained in affiliation names. The words in field-of-studies can be seen as sharing the same language with paper contents and most venue names also contain informative concept words such as “Machine Learning” or “High Energy”. This is not always true for affiliation names. For universities like “Harvard University” or “University of Oxford”, their researchers could focus on multiple unrelated domains which are hard for language models to capture. For companies and research institutes, some may focus on a single domain but it is not necessary to have such descriptions in their names, which also confuses the pre-training language model.

Discussion for Entity Probability In Equation 2, we use the sum of log probabilities of all tokens to calculate the entity log probability. This method seems to be unfair for entities with longer lengths as the log probability for each token is always negative. However, for MLM-based models, the encoding process not only encodes “[MASK]” tokens but also captures the length of the masked entity and each token’s position. Therefore, if the pre-training corpus has fewer long entities than short entities, in the decoding process, the decoded tokens in a long entity will generally receive higher probability, compared to the ones in a short entity.

Even so, the sum of log probabilities is still not necessary to be the best choice depending on the entity distribution in the pre-training corpus. We conduct a simple experiment to test different average methods. We reform the calculation of entity log probability in Equation 2 as $\frac{1}{L^{\alpha}} \sum_{1 \leq k \leq l} \log P(w_{i_k} | C, w_{i_1}, w_{i_2}, ..., w_{i_{k-1}})$, where L denotes the length of target entity. When $\alpha = 0$, this equation degrades to the summation version, which is used in previous tasks. When $\alpha = 1$, this equation degrades to the average version.

We compare different averaging methods by using various α and test their performance on the zero-shot inference tasks. We select the input features with the best performance according to Table 2. For SciBERT, we use both abstract and prompt word for FOS and affiliation inference. We do not use the prompt word for venue inference. For OAG-BERT, we only use abstract as the prompt word does not work well. The results in Table 8 show that for the most time, using the summation strategy outperforms the average strategy significantly. The simple average ($\alpha = 1$) appears to be the worst choice. However, for some situations, a moderate average ($\alpha = 0.5$) might be beneficial.

Table 8: The results for using different average methods while calculating entity log probabilities. Hit@1 and MRR are reported.

Method	$\alpha = 0$	$\alpha = 0.5$	$\alpha = 1$
SciBERT			
FOS	**35.33%, 0.52**	32.07%, 0.51	14.85%, 0.36
Venue	18.00%, 0.32	**19.30%, 0.33**	7.07%, 0.23
Affiliation	**12.40%, 0.25**	10.83%, 0.23	9.23%, 0.21
OAGBERT			
FOS	**49.59%, 0.67**	48.08%, 0.66	45.36%, 0.63
Venue	**39.00%, 0.57**	38.20%, 0.57	36.13%, 0.55
Affiliation	**21.67%, 0.38**	19.90%, 0.36	16.47%, 0.31

Discussion for Decoding Order In our designed decoding process, we do not strictly follow the left-to-right order as used in classical decoder models. The main reason is that for encoder-based BERT model, the decoding for each masked token relies on all bidirectional context information, rather than only prior words. We compare the performance of using left-to-right decoding and out-of-order decoding in Table 9.

The results show that for FOS, there is no significant difference between two decoding orders, since the candidate FOS only have one or two tokens inside. As for venue and affiliation, it turns out that the out-of-order decoding generally performs much better than left-to-right decoding, except when OAG-BERT is using abstract where differences are relatively small as well. We also present the results for models using left-to-right decoding and prompt words in Table 9, which indicates that the left-to-right decoding will sometimes undermine the effectiveness of prompt words significantly, especially for OAG-BERT.

A.2 Supervised Classification

In terms of training, we use the slanted triangular scheduler to adjust the learning rate dynamically and the AdamW optimizer with a maximal learning rate at 2e-5. We run the fine-tuning process for 5 epochs with 10% of the training steps used for warm-up. For each model and each task setting, the averaged accuracy (Hit@1) and standard deviations for 5 runs with different random seeds are reported in Table 4. The number of samples in the classification datasets is shown in Table 11.

Table 9: The results for using left-to-right decoding and out-of-order decoding order. Hit@1 and MRR are reported. Results with difference larger than 1% Hit@1 were bolded.

Method	FOS		Venue		Affiliation	
	Hit@1	MRR	Hit@1	MRR	Hit@1	MRR
SciBERT						
Left-to-Right	20.05%	0.37	8.40%	0.20	6.90%	0.18
Out-of-Order	19.93%	0.37	**9.87%**	**0.22**	6.93%	0.19
SciBERT *+prompt*						
Left-to-Right	29.65%	0.47	9.57%	0.21	8.03%	0.20
Out-of-Order	29.59%	0.47	10.03%	0.21	8.00%	0.20
SciBERT *+abstract*						
Left-to-Right	25.67%	0.43	11.43%	0.24	7.63%	0.19
Out-of-Order	25.66%	0.43	**18.00%**	**0.32**	**10.33%**	**0.22**
SciBERT *+both*						
Left-to-Right	35.21%	0.52	**11.17%**	**0.24**	11.47%	0.23
Out-of-Order	35.33%	0.52	9.83%	0.22	12.40%	0.25
OAG-BERT						
Left-to-Right	34.94%	0.53	11.33%	0.24	5.47%	0.17
Out-of-Order	34.36%	0.51	**21.00%**	**0.37**	**11.03%**	**0.24**
OAG-BERT *+prompt*						
Left-to-Right	37.84%	0.56	12.53%	0.26	5.50%	0.17
Out-of-Order	37.33%	0.55	**22.67%**	**0.39**,	**11.77%**	**0.25**
OAG-BERT *+abstract*						
Left-to-Right	49.75%	0.67	**40.50%**	**0.59**	21.93%	0.38
Out-of-Order	49.59%	0.67	39.00%	0.57	21.67%	0.38
OAG-BERT *+both*						
Left-to-Right	49.83%	0.67	22.17%	0.38	6.80%	0.19
Out-of-Order	49.51%	0.67	**38.47%**	**0.57**	**21.53%**	**0.38**

A.3 NLP Tasks

Task Description Among all 15 NLP tasks, 9 tasks concentrate on the field of Biology and Medicine (Bio). Another 4 tasks use paper samples from computer science domain (CS). The rest two tasks involve a mixture of multi-domain data (Multi).

Tasks including NER and PICO require models to make predictions on each token and identify which tokens are part of entities. Some datasets like BC5CDR [26] only need span range identification while other datasets like EBM-NLP [33] also need entity type recognition. For sequence token classification tasks, a Conditional Random Field (CRF) layer is added on top of token outputs from the pre-training model, to better capture the dependencies between sequence labels. The DEP task [20] also uses the token outputs from the pre-training model. The token embeddings, produced by the pre-training model, are fed to a biaffine matrix attention block and used to make further predictions on dependency arc type and direction. The REL and CLS tasks are sequence prediction tasks. The model only needs to make one prediction on the whole sequence. For example, in Paper Field prediction task [38], the model accepts paper title as inputs and output the research fields of that paper. The REL tasks [24, 31], although not directly asking the label of the input sequence, can be reformed into sequence prediction as well. In this type of tasks, the model makes predictions for the entity relation types by categorizing the whole sequence, where the focused entity pairs are encapsulated with special tokens. The SSC tasks are multi-sequence prediction tasks. Given a list of sentences such as abstract, the model needs to predict the functionality for each internal sentence. These tasks always involve long sequences and also benefits from using CRF layer on top of the sentence embeddings.

Table 10: A full list of used candidates in zero-shot inference tasks and supervised classification tasks.

FOS: Art, Biology, Business, Chemistry, Computer science, Economics, Engineering, Environmental science, Geography, Geology, History, Materials science, Mathematics, Medicine, Philosophy, Physics, Political science, Psychology, Sociology

Venue: Arxiv: algebraic geometry, Arxiv: analysis of pdes, Arxiv: astrophysics, Arxiv: classical analysis and odes, Arxiv: combinatorics, Arxiv: computer vision and pattern recognition, Arxiv: differential geometry, Arxiv: dynamical systems, Arxiv: functional analysis, Arxiv: general physics, Arxiv: general relativity and quantum cosmology, Arxiv: geometric topology, Arxiv: group theory, Arxiv: high energy physics - experiment, Arxiv: high energy physics - phenomenology, Arxiv: high energy physics - theory, Arxiv: learning, Arxiv: materials science, Arxiv: mathematical physics, Arxiv: mesoscale and nanoscale physics, Arxiv: nuclear theory, Arxiv: number theory, Arxiv: numerical analysis, Arxiv: optimization and control, Arxiv: probability, Arxiv: quantum physics, Arxiv: representation theory, Arxiv: rings and algebras, Arxiv: statistical mechanics, Arxiv: strongly correlated electrons

Affiliation: Al azhar university, Bell labs, Carnegie mellon university, Centers for disease control and prevention, Chinese academy of sciences, Electric power research institute, Fudan university, Gunadarma university, Harvard university, Ibm, Intel, Islamic azad university, Katholieke universiteit leuven, Ludwig maximilian university of munich, Max planck society, Mayo clinic, Moscow state university, National scientific and technical research council, Peking university, Renmin university of china, Russian academy of sciences, Siemens, Stanford university, Sun yat sen university, Tohoku university, Tsinghua university, University of california berkeley, University of cambridge, University of oxford, University of paris

Table 11: The sizes for datasets used in supervised classification tasks.

Task	Categories	Train	Validation	Test
FOS	19	152000	19000	19000
Venue	30	24000	3000	3000
Affiliation	30	24000	3000	3000

Table 12: Details for the CS heterogeneous graph used in the link prediction.

	Papers	Authors	FOS
Nodes	544244	510189	45717
1116163	Venues	Affiliations	
	6934	9079	
	#Paper-Author	#Paper-FOS	#Paper-Venue
#Edges	1862305	2406363	551960
6389083	#Author-Affiliation	#Paper-Paper	#FOS-FOS
	519268	992763	56424

Evaluation Metrics We use the same evaluation metrics with the SciBERT [2] paper and the Sequential-Sentence-Classification [5] paper. For NER and PICO tasks, we compare the span-level and token-level macro F1 scores respectively, except using micro-F1 for ChemProt [24]. For REL, CLS, and SSC tasks, we compare sentence-level macro F1 scores. For the DEP task, we compare LAS (labeled attachment score) and UAS (unlabeled attachment score).

Hype-parameters In SciBERT, the authors claimed that the best results for most downstream tasks were produced by fine-tuning 2 or 4 epochs and using 2e-5 learning rate after searching between 1 to 4 epochs with a maximum learning of 1e-5, 2e-5, 3e-5, 5e-5, as stated in [30]. In our experiments, we follow the same settings and select the optimal hyper-parameters on validation sets and report the corresponding test sets results.

Table 13: The performance of vanilla OAG-BERT with and without training on 512-token samples. All results in this table were produced by fine-tuning with 2 epochs and 2e-5 learning rates.

Task	Dataset	Vanilla OAG-BERT		Gain
		w/o 512	*w/ 512*	
NER	BC5CDR	$89.62^{.16}$	$89.33^{.12}$	-0.29
	NCBI-disease	$87.63^{.62}$	$87.92^{1.08}$	+0.29
	SciERC	$67.64^{.52}$	$67.19^{.34}$	-0.45
REL	ChemProt	$77.50^{1.99}$	$77.99^{2.50}$	+0.49
	SciERC	$69.87^{1.51}$	$69.88^{.77}$	+0.01
SSC	NICTA-piboso	$77.62^{.87}$	$80.01^{.24}$	**+2.39**
	CSAbstruct	$72.65^{.40}$	$82.30^{.47}$	**+9.65**

Pre-training on 512-token samples During fine-tuning on NLP tasks, we also observe that the pre-training on inputs with 512 tokens is essential for the SSC tasks with up to 10% performance boost, which is much larger than the performance boost for other types of tasks as shown in Table 13. It is because the SSC tasks require the model to comprehend multiple sentences in a long paragraph rather than a single sentence in other tasks.