2021 International Conference on Data Mining Workshops (ICDMW) | 978-1-6654-2427-1/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICDMW53433.2021.00013

CoBERT: Scientific Collaboration Prediction via Sequential Recommendation

Tobias Koopmann, Konstantin Kobs, Konstantin Herud, Andreas Hotho
Chair of Data Science
University of Würzburg
{koopmann,kobs,konstantin.herud,hotho}@informatik.uni-wuerzburg.de

***Abstract*—Collaborations are an important factor for scientific success, as the joint work leads to results individual scientists cannot easily reach. Recommending collaborations automatically can alleviate the time consuming and tedious search for potential collaborators. Usually, such recommendation systems rely on graph structures modeling co-authorship of papers and content-based relations such as similar paper keywords. Models are then trained to estimate the probability of links between certain authors in these graphs.**

In this paper, we argue that the order of papers is crucial for reliably predicting future collaborations, which is not considered by graph-based recommendation systems. We thus propose to reformulate the task of collaboration recommendation as a sequential recommendation task. Here, we aim to predict the next co-author in a chronologically sorted sequence of an author's collaborators. We introduce CoBERT, a BERT4Rec inspired model, that predicts the sequence's next co-author and thus a potential collaborator. Since the order of co-authors of a single paper is not that important compared to the overall paper order, we leverage positional embeddings encoding paper positions instead of co-author positions in the sequence. Additionally, we inject content features about every paper and their co-authors. We evaluate CoBERT on two datasets consisting of papers from the field of Artificial Intelligence and the journal PlosOne. We show that CoBERT can outperform graph-based methods and BERT4Rec when predicting the co-authors of the next paper. We make our code and data available.

***Index Terms*—sequential recommendation, bibliometric research, co-author prediction**

I. Introduction

Bibliometric research has shown that collaboration between scientific researchers is a major factor for the success of research projects, since collaborators with different expertise can contribute ideas [1]. Due to the large pool of possible collaborators in many scientific fields, identifying potential collaborators for new research projects becomes very difficult. Different and partially hidden factors can play a vital role for successful cooperation. Thus, scientific collaboration recommendation systems analyze large corpora of publications to identify potential collaboration partners depending on previous collaborations or thematic fit.

Bibliometric research determined different facets of reasons, why researcher collaborate, for example similar professional knowledge (cognitive proximity), previous acquaintances (social proximity) or geographic closeness (geographic proximity) [1]. These proximities can change over time, as researchers adapt their research topics, get to know new possible collaborators, or change their current affiliation.

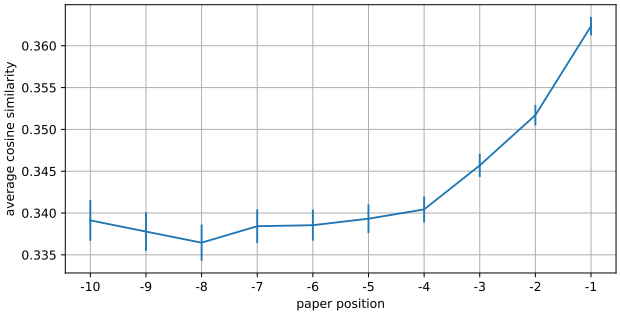

Fig. 1: Mean cosine similarity (and standard error) between authors' latest and preceding papers (embedded using SentenceBERT [4]). The older the paper, the lower the similarity. Thus, CoBERT takes the order of papers into account.

Current co-author recommendation models rely on collaboration graphs, which model collaboration behavior of researchers using co-authorships [2], [3]. In these graphs, each node represents an author and edges connect nodes if the respective authors have collaborated on a publication. Features extracted from these graphs as well as content based features are then used to predict missing links, i.e. edges that a model would classify as being present in the graph but are not. Missing links are interpreted as potential collaboration partners and thus recommended to the respective authors. Given the proximities introduced by Boschma [1], existing links in the graph approximate social proximity and content based features aim to represent cognitive proximity. Nevertheless, graphs do not incorporate the temporal aspects of the research journey of an author. We argue that taking the chronological order of collaborations into account is important when predicting future collaborators, as it reveals changing proximities that have an impact on collaboration decisions. We showcase this by visualizing the mean cosine similarities between authors' latest and preceding papers in Figure 1 (using the AI dataset with $n = 5$, c.f. Section IV-A). More recent papers have higher similarity, thus should be better indicators for cognitive proximity than older papers.

Based on this observation, we propose to reformulate the

978-1-6654-2427-1/21/$31.00 ©2021 IEEE
DOI 10.1109/ICDMW53433.2021.00013

scientific collaboration prediction task as a sequential recommendation Wang, Hu, Wang, *et al.* and introduce CoBERT, a sequential recommendation model for this exact task. An input sequence consists of all co-authors an author has collaborated with in the past, sorted by the publishing date of their co-authored paper. In this way, the research journey of an author can be captured, since the order of papers can show trends in collaborations more easily. Furthermore we enrich each sequence item with content-based features such as paper representations or features describing the co-authors' research interests. To the best of our knowledge, we are the first to adapt a sequential recommendation model to the task of scientific collaboration prediction. We then propose three modifications that substantially help the model to better predict collaborations:

1) We adapt the positional encoding for each item to our setting. Since multiple co-authors can appear on the same paper, we introduce a positional encoding *per paper* and add the same encoding to each co-author of the respective paper. This makes the order of co-authors per paper irrelevant to the model.
2) We add content-based features of each paper to the sequence. For this, we compute semantic paper content embeddings for the abstract and keywords of each paper using Sentence-BERT [4] and add them to the corresponding sequence items [6].
3) Instead of learning co-author embeddings from scratch, we initialize them with vectors representing the co-author's research interest. We compute them by aggregating the paper content embeddings for all papers of each co-author.

We extensively evaluate CoBERT in different scenarios, finding that CoBERT can substantially improve collaboration prediction quality compared to weak and strong baselines, including graph based methods. An ablation study shows that the model benefits from our proposed modifications.

For our experiments, we utilize two datasets: A medium-sized dataset consisting of papers from **Artificial Intelligence (AI)** conferences and a large-scale dataset containing all papers published in the **PlosOne** journal. To examine the influence of small input lengths on our model, we create two versions of each dataset by filtering authors with a small number of co-authors. This yields two datasets, one with *more but shorter sequences*, and hence giving less context per author, and a second one with *fewer but longer sequences* and therefore more context information.

We explore two tasks, defining different ways to recommend co-authors: **New Collaborator Prediction** identifies fitting co-authors for the next paper given an author's previous papers and their corresponding co-authors, while only recommending co-authors the author has not yet collaborated with. This resembles the usual task in collaboration recommendation settings, i.e. predicting missing links in the collaboration graph is equivalent to connecting authors that have not appeared on a paper together [7], [8]. This also is a realistic use case for a collaboration recommendation system, e.g. when researchers want to find potential new collaborators for a new project. In reality, however, authors usually repeatably collaborate with the same co-authors [9]. Thus, we introduce the **Any Collaborator Prediction** task, in which the model has to predict the next co-author, regardless of previous collaborations. In this task, the prediction can contain co-authors that already appeared in the input sequence. Predicting new collaborators is the more difficult task since the model cannot just predict a co-author from the input but has to extrapolate to unseen potential co-authors based on similar authors or meta-information.

Our contributions are as follows:

1) We propose to interpret the collaboration prediction task as a sequential recommendation task and propose our model CoBERT to solve this task.
2) We do extensive experimentation to evaluate CoBERT and investigate the effects of each of the proposed modifications.
3) We make our code and data available.[1]

The remainder of this work is structured as follows. First, Section II covers related work. Section III introduces and describes the methodology of CoBERT. In Section IV, we explain the experimental setup and results as well as provide an ablation study for CoBERT's components. We finish with a conclusion in Section V.

II. Related Work

Most approaches for the task of co-author recommendation build a collaboration graph from the training data, where each author is a node and two nodes are connected by an edge for each paper the authors have collaborated on. Recommending new collaborators for an author then resembles a link prediction task, i.e. estimating the probability that two nodes are connected in the collaboration graph [2], [3]. This prediction usually relies on graph features [7], [8] or on latent representations extracted from the graph structure [2], [10]. For larger graphs it becomes computationally expensive to calculate node features, for example personalized page-rank [7]. On the other hand, latent node and graph representations can be generated using the skip-gram approach on biased random walks Node2Vec [2] and are easier to compute even for large graph structures. Additionally, recent work enhances these graph representations with meta-information, for example by incorporating textual features [11], [12]. Finally, some approaches do not use deep learning to solve this task. Daud, Ahmad, Malik, *et al.* [13] are exploring approaches as Markov models, regression trees, or Bayesian networks as models to tackle co-author prediction.

We interpret co-author recommendation as a sequential recommendation task where the goal is to predict the next item in a sequence. In sequential recommendation, such sequences usually consist of click streams or rating histories [14], [15]. To model sequences, different architectures such as RNNs [16], CNNs [17], recurrent CNNs [18], or self-attention networks [19] have been explored. Sun, Liu,

[1]Will be made available upon acceptance.

Wu, *et al.* have proposed BERT4Rec, an adaption of the bidirectional transformer model BERT [20] to the sequential recommendation task [14]. While per default, sequential recommendation models only make predictions solely based on the sequence itself, giving them access to content based information can be beneficial. Thus, multiple works have explored ways to encode additional information, e.g. text using 3D convolutions [21] or multi-modal data using multiple RNNs encoders [22]. Recently, BERT4Rec was extended to encode content information by adding item keyword embeddings to the input embeddings [6].

III. CoBERT

This section describes CoBERT, our proposed model for scientific collaboration prediction. Since one of our main contribution is to redefine the task of co-author prediction as a sequential task, we resort to sequential recommendation models. For this, we adapt BERT4Rec, a transformer based BERT model introduced for sequence-based recommendation. As input we interpret the collaborators of an author as a sequence, sorted by the respective paper's time of publication. We argue that by modeling the collaboration and paper history, the model can consider shifts in content and co-authors over time and thus recommend future collaborations according to observed trends. To model such information, we alter the input embeddings to the model by additionally using the content of each paper (abstract and keywords) and their co-authors. In the following, we introduce the notation, then briefly describe the sequential model BERT4Rec, and explain how to apply it to the co-author prediction task. The, we introduce each proposed modification that makes up CoBERT. A schematic overview of the final CoBERT model is depicted in Figure 2.

A. Notation

Let $\mathcal{A} = \{a_1, a_2, \ldots, a_{|\mathcal{A}|}\}$ be a set of authors, $\mathcal{P} = \{p_1, p_2, \ldots, p_{|\mathcal{P}|}\}$ a set of scientific papers. Further, $\mathcal{C}_p$ is the list of authors of paper p and $\mathcal{K}_a$ is a list of papers author a has collaborated on, sorted by their publication date. Then, the list $\mathcal{S}_a = (a_i \mid a_i \in \mathcal{C}_p \setminus a \quad \forall p \in \mathcal{K}_a)$ contains all co-authors of an author a, sorted by the respective paper's publication date. Note that this sequence can contain the same co-author repeatedly. Given such a sequence $\mathcal{S}_a$, the task is to predict the probability of the next author in the sequence:

$$P(\mathcal{S}_{a;|\mathcal{S}_a|+1} | \mathcal{S}_{a;1}, \ldots, \mathcal{S}_{a;|\mathcal{S}_a|}) \,. \tag{1}$$

B. BERT4Rec

To process the sequential data, we adapt the sequential recommendation model BERT4Rec, which predicts the next item in a sequence using a bidirectional self-attention architecture [14]. In the following, we explain the general architecture of this model such that it fits the collaboration prediction task. BERT4Rec's basic components are the input, the embedding, the transformer, and projection layers. Since we do not change the transformer layers $\mathcal{T} = \{t_1, \ldots, t_L\}$ and projection layers $proj$, we refer to [14] for details about these components.

In our setting, the co-author sequence S_a for an author a is the input to the model. Each co-author in this sequence is represented as an embedding vector $\mathbf{a}_i$ for sequence position i. Each embedding is summed with the corresponding positional embedding $\mathbf{p}_i$ encoding the sequence position i to allow the transformer to take the order of inputs into account [20]. This gives input representations $\mathbf{h}^0 = \mathbf{c} + \mathbf{p}$, which are fed through the transformer layers $t \in \mathcal{T}$. As in the BERT4Rec training procedure, we replace randomly selected sequence items with the mask token `[mask]` during training (see Section III-G). The network is trained to predict the masked item from the final transformer hidden state t_L at the masked item's position using a projection layer $proj$ and the categorical cross entropy loss function. In this way, the network learns to represent items using other items in the sequence, picking up trends and patterns from the training data. At inference time, a mask token is appended to the full sequence, letting the model predict the probability for the next item in the sequence $P(\mathcal{S}_{a;|\mathcal{S}_a|+1} | \mathcal{S}_{a;1}, \ldots, \mathcal{S}_{a;|\mathcal{S}_a|})$. All possible items are ranked by the magnitude of their projection layer output.

C. Modification 1: Positional Embedding per Paper (PPE)

The input sequence for an author $\mathcal{S}_a$ consists of co-authors they have worked with sorted by the time the corresponding paper was published. BERT4Rec poses the problem that each item is provided with a positional embedding that encodes the absolute position of the item in the sequence. Since the exact order of co-authors is not as relevant as the order of the respective publication $\mathcal{K}_a$ for an author, we assign the same positional embedding to all co-authors of a paper in the sequence. For example, if there are three co-authors for the first paper and two authors for the second paper in the author's sequence, the positional encoding for the first three items in the sequence is the same, encoding position one, and the positional encoding for the forth and fifth sequence item encodes position two. This gives the model a notion of order of the papers but does not enforce an order of co-authors. Since transformers are insensitive to item positions without a positional encoding [20], the model will not take the order of co-authors for one paper into account but only the paper order. To better work with the sequence item indices, we define a function f that maps a sequence position i to the corresponding paper position regarding this sequence. In the example above, $f(1) = f(2) = f(3) = 1$ and $f(4) = f(5) = 2$.

D. Modification 2: Paper Content Embedding (PCE)

Besides order, we argue that cognitive proximity [1] is an important factor in estimating future co-authors. Cognitive proximity represents the similarity of authors with respect to their prior professional knowledge. We represent this knowledge by leveraging content information of papers and thus get a thematic classification for each author. Co-authors are then selected according to topics, and authors from papers with different topics are less likely to be a suitable match for the next paper. We propose to take information about

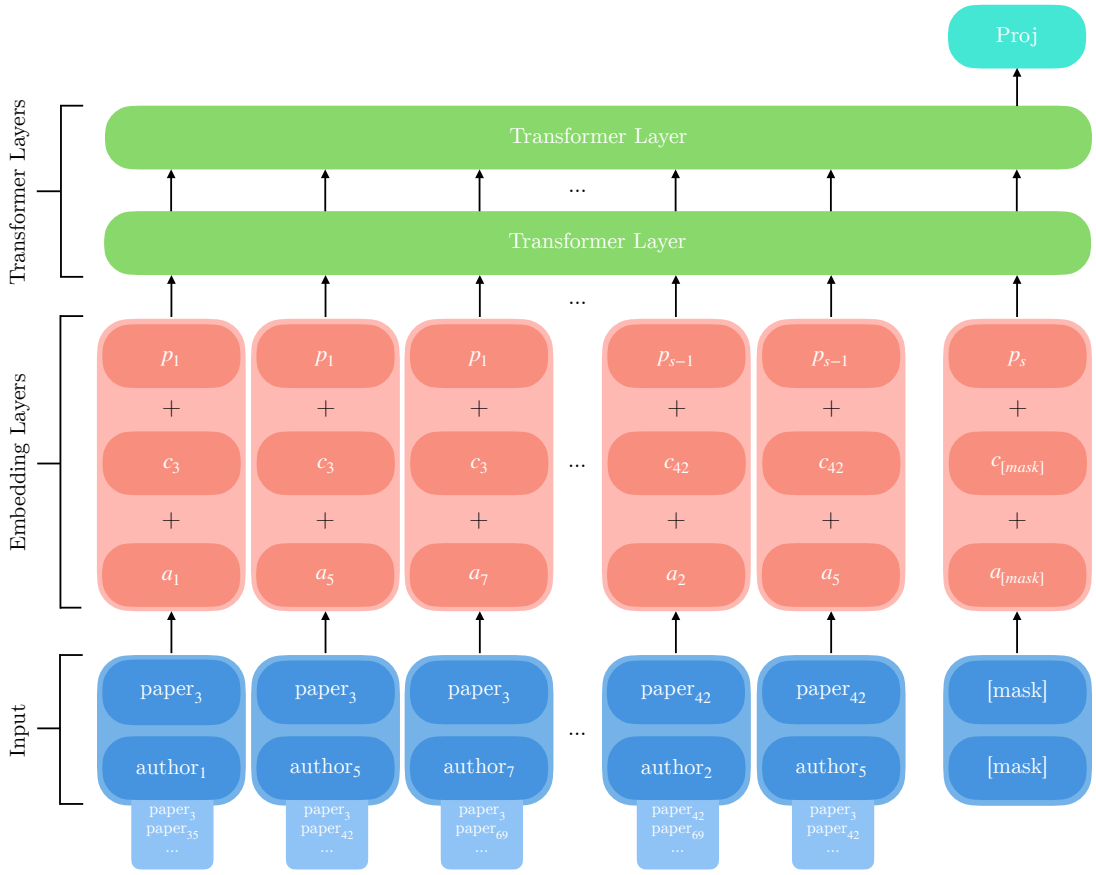

Fig. 2: Overview of CoBERT. The input to the model is a sequence of paper with the respective co-authors, with which the current author has collaborated with, sorted by publication date. For each item, we additionally leverage a representation of abstract and keywords of the corresponding paper. We then compute a positional encoding per paper p_i as well as embeddings to describe the content of the paper c_i. Finally we represent the co-author's previous works a_i by average pooling all paper embeddings p of paper written by the current author. These embeddings are then summed and used as input to transformer layers. Appending mask tokens to the sequence allows the model to predict the next co-author in the sequence, using the output representation from the transformer and a fully-connected projection layer (Proj).

papers into account by enriching the sequence inputs with embeddings created from the corresponding papers the co-authors are part of. For this, we leverage SentenceBERT [4], which is designed to embed documents of short length[2]. We concatenate the keywords and abstracts of each paper and feed it through SentenceBERT to obtain representations, which we call Paper Content Embeddings (PCE). Similarly to the positional embedding, this representation $\mathbf{c}_p$ is repeated for each co-author of the encoded paper p. The function g maps the sequence position i to the corresponding paper.

E. Modification 3: Co-Author Content Embedding (ACE)

Up until now, each sequence item encodes a co-author as a one-hot encoded vector, which then gets represented by a lower dimensional dense vector in the embedding layer. We propose to use a smarter initialization for the co-author embedding by computing the mean PCE over all papers of a co-author: $\mathbf{a}_a = \frac{\sum_{p \in \mathcal{K}_a} \mathbf{c}_p}{|\mathcal{K}_a|}$. In this way, co-authors are already represented by their research contents, bringing authors with similar research interests closer together in embedding space. A function h maps the sequence item position i to the corresponding co-author, so multiple occurrences of the same co-author are mapped to the same co-author content embedding.

F. Combining All Embeddings

Inspired by Fischer, Zoller, Dallmann, *et al.* [6] who add keyword embeddings to sequence items in BERT4Rec, we obtain the initial representation that is fed into the transformer layers by summing all separate embeddings:

$$\mathbf{h}_i^0 = \underbrace{\mathbf{p}_{f(i)}}_{\substack{\text{Positional Encoding} \\ \text{per Paper}}} + \underbrace{\mathbf{c}_{g(i)}}_{\substack{\text{Paper Content} \\ \text{Embedding}}} + \underbrace{\mathbf{a}_{h(i)}}_{\substack{\text{Co-Author Content} \\ \text{Embedding}}} . \quad (2)$$

[2]We use https://huggingface.co/sentence-transformers without fine tuning.

Here, all embeddings are weighted equally. We have experimented with variable weights per summand that are optimized jointly with the neural network parameters, but this has led to low performance. Details to this and other failed approaches can be found in Appendix B.

Overall, the changes proposed above define our model **CoBERT** (c.f. Figure 2).

G. Training

The training of CoBERT mostly follows the training procedure of BERT4Rec [14]. In general, the used cloze task masks items in the sequence with a `[mask]` token. In $5\,\%$ of all cases, only the last item is masked. This scenario approximates the actual task of predicting the next co-authors. In the remaining cases, each item is selected for the masking process with a probability of $20\,\%$, while a maximum selection size of half the sequence length should not be exceeded. The masking process inserts the `[mask]` token for the item with an $80\,\%$ chance. For $10\,\%$, this co-author is replaced with a randomly chosen co-author. Else, no change is made to the item. This altered sequence is then fed into the model, which has to learn to predict the masked items from their context. For this, we use the hidden representations of the transformer layers at the positions of all altered tokens and use a projection layer to predict the correct co-author with a softmax activation function. In this way, the network has to (1) fill in the blanks if an item is masked, (2) learn to correct mistakes if the item is replaced with another item, and (3) be aware of correct items.

During inference, a `[mask]` token is appended to the whole sequence. The model then output a probability distribution over all possible co-authors for this token. Co-authors are then ranked using their corresponding probability.

IV. Experiments

In this section we describe the experiments used to evaluate CoBERT as well as their results. We run experiments on two datasets with two different minimal sequence lengths and define two tasks to better investigate the effects of different influences on the performance of CoBERT and other baselines. We use PyTorch [23] for our experiments.

A. Datasets

We use two datasets with publications from Artificial Intelligence (AI) research and the interdisciplinary journal PlosOne.

AI Dataset: For this, we utilize a published dataset[3] consisting of meta-data from publications published at AI conferences as categorized by Kersting, Peters, and Rothkopf [24]. Each paper contains information like co-authors, conference, year, keywords, and abstracts. We can thus model and predict the paper sequences for authors from these conferences.

PlosOne: We build a second dataset that consists of information about papers published in the journal PlosOne. We extract all information from Semantic Scholar Open Research Corpus[4] [25], i.e. co-authors, year, abstract and keywords. While the AI dataset contains papers with a thematic focus on Artificial Intelligence, PlosOne is an interdisciplinary journal, thus containing papers from a large variety of research areas. This could make recommending collaborations easier for PlosOne, since researchers usually collaborate in their own sub-domain without much overlap to other research areas. On the other hand, this dataset has much more authors than the AI dataset, making the prediction of co-authors more difficult.

1) Pre-Processing: From both datasets, we can generate co-author sequences for an author in order to predict the next item in the sequence using CoBERT. To give the model some context, we remove authors with co-author sequences shorter than n. We set n to 5 or 10 to investigate the effects of giving the model more or less context for each instance. After this step, we obtain two datasets with the properties displayed in Table I. While with $n = 5$, the datasets consist of more papers and authors overall, which can be useful during training, the average number of co-authors per author is smaller and thus the model input sequence is shorter. Due to the shorter context per author, we thus consider the dataset with $n = 5$ to be a more difficult dataset.

2) Data Split: We split each co-author sequence into training, validation, and test sequences. Since we are the first to model co-author recommendation as sequential task, we introduce a new data splitting procedure. It is different for the two tasks, however, both rely on the chronologically sorted co-author sequence.

Any Collaborator Prediction: We put all co-authors of the latest paper into the test set and one co-author from the second latest paper into the validation set. When evaluating, the ranking of these co-authors is then measured.

New Collaborator Prediction: We select one co-author that only occurs once in the sequence to select only novel and no repeating collaborations. For identifying this co-author, we begin at the end of the sequence, such that the latest collaborations are preferably selected. We delete this co-author from the sequence and repeat the process to find a co-author for the validation set. By selecting test co-authors that only have collaborated once with the sequence's author, this prevents much shorter training sequences. On the other hand, this makes the task more difficult to solve, since one-time collaborations in the past may be due to arbitrary influences and not similar research interests. While it is possible that some author sequences do not have single occurrences of co-authors, we find that approximately $90\,\%$ to $95\,\%$ of all sequences do. We thus only use these sequences for validation and testing. In order to predict only new collaborators, we remove past collaborators from CoBERT's predicted ranking.

B. Baselines

We compare CoBERT to different baselines. First, a *majority* baseline ranks authors based the length of $\mathcal{K}_a$, i.e. their number of occurrences in the training set. This baseline predicts this ranking for each author, which is used to calculate our metrics. Additionally, we implement a stronger majority baseline (called *Majority (author)*) that ranks co-authors for

[3]https://zenodo.org/record/3693604#.YIfQDi0RphE

[4]Dump from 2019-01-31, since it is the last dump containing keywords.

TABLE I: Dataset characteristics. n represents the minimal sequence length in order to be considered in the dataset.

	AI		PlosOne	
	$\mathbf{n=5}$	$\mathbf{n=10}$	$\mathbf{n=5}$	$\mathbf{n=10}$
authors/sequences	65,541	35,113	205,544	132,003
papers	175,237	138,344	135,896	95,646
avg. co-authors	48.89	69.04	47.77	59.60
avg. unique co-authors	8.61	12.32	3.59	4.03
avg. sequence length	24.45	34.52	23.88	29.80
sequence length > 5	94.06 %	100.00 %	97.47 %	100.00 %
sequence length > 10	57.89 %	91.56 %	72.09 %	93.40 %
sequence length > 15	39.88 %	63.75 %	48.85 %	65.06 %

each author individually. Given a sequence $\mathcal{S}_{\dashv}$ of an author, we calculate a ranking based on the occurrence within this sequence. This baseline does only predict co-authors, with whom the author already collaborated, which does not allow extrapolation to new authors. Hence this baseline is inapplicable for the new collaborator prediction task.

Second, we compare CoBERT, that is built on the idea of sequential recommendation, to a more common approach utilizing *collaboration graphs* [2], [3]. The graph is built from the training data: Nodes represent authors and edges represent paper collaborations between authors. State-of-the-art co-author prediction models leverage graph features [7], [8], such as a personalized page rank. For our datasets, the collaboration graph is too large to efficiently compute such graph features. For example, personalized page rank has a computational time of $\mathcal{O}(n(n+m))$, where n is the number of nodes and m is the number of edges. We thus use efficiently computable node embeddings that are computed from the graph using the Deepwalk approach [26]. Node embeddings capture the author's graph neighborhood and thus collaborations. In order to predict links between two authors, we element-wise add their node embeddings (following Grover and Leskovec [2]) and use the resulting representation as input to a multi-layer perceptron classifier with one hidden layer (size 100). As output, the network gives a probability given the pair of authors, which is used to generate a ranking over all possible co-authors. This ranking is returned as the prediction.

Finally, to show that with CoBERT we propose sensible changes to the BERT4Rec architecture, we also use BERT4Rec as a baseline in our experiments. BERT4Rec uses a positional encoding per co-author and does not add content based features about the paper and co-author to the input sequence. The training procedure stays the same as with CoBERT, in order to compare both approaches in a fair setting.

C. Hyperparameters

For BERT4Rec and CoBERT, we adopt the default BERT4Rec hyperparameters: The network consists of two transformer layers with eight heads each and a hidden size of 768. We use Adam [27] with a learning rate of 0.001 and apply early stopping [28] with a patience of five epochs. We choose a batch size of 16.

D. Evaluation Metrics

We evaluate the resulting models using the metrics *Hitrate (Hit@{1,5,10})* and *Normalized Discounted Cummulative Gain (NDGC@{5,10})*. As commonly used in recommendation settings and following BERT4Rec, we use negative sampling in our evaluation with $n = 100$ in order to handle the large co-author space [14].

E. Results

We train each approach five times and report mean metrics in Tables II and III. While the AI dataset is smaller but more focused, the PlosOne dataset is larger but thematically more variable. For each dataset, we have defined two minimal sequence lengths, resulting in potentially shorter but more co-author sequences ($n = 5$) as well as longer but fewer sequences ($n = 10$). Additionally, we evaluate all methods on two tasks: Predicting any co-author regardless of past collaborations (Any Collab.) and predicting new co-authors that have not appeared in the input sequence (New Collab.). Overall, our proposed model CoBERT outperforms all baselines most of the times. Only for the PlosOne dataset with $n = 5$ and predicting any collaborator, the majority baseline per author performs better on Hit@10. This shows for this dataset, authors usually collaborate with the same co-authors in the future. However, a lower NDCG@10 shows that CoBERT ranks the correct co-authors higher than the majority per author baseline. We report our findings for all of the different variations in dataset and tasks in the following.

1) Dataset: Given the results, the performance of all models drops for the PlosOne dataset. This indicates that the PlosOne dataset is more challenging. We hypothesize that this is due to the larger pool of possible co-authors (cf. Table I). In addition, there are more authors per paper in the PlosOne dataset. When removing the co-authors of the last paper for the test data, the input sequence becomes significantly shorter, making the task more difficult.

2) Sequence Length: Training and evaluating CoBERT on longer sequences tends to result in higher performance, even though there are fewer sequences available (cf. Table I). This might be due to the fact that it is easier for the model — as it is for humans — to identify useful patterns in longer sequences.

3) Task: As expected, the prediction of a co-author that does not appear in the input sequence (New Collab.) is a more difficult task than allowing the prediction of past collaborators (Any Collab.). While the models are still able to extrapolate to new co-authors, the performance drops substantially on all metrics for the AI dataset. On the PlosOne dataset, smaller performance drops can be observed. Since PlosOne is an interdisciplinary journal, it might be easier for CoBERT to identify novel co-authors by recommending co-authors from thematically fitting subgroups.

F. Ablation Study

Our results show that CoBERT achieves better performance than the methods we compared it to. We now want to examine the influence of each of our proposed modifications to the

TABLE II: Results on the **AI** dataset. Given are the evaluation metrics' means over five runs. All values are in percent. Best values are written in bold. The majority baseline per author cannot be evaluated for the New Collaborator Prediction task, since it does not predict any new collaborators.

		$\mathbf{n = 5}$					$\mathbf{n = 10}$				
Task	**Model**	**Hit@1**	**Hit@5**	**Hit@10**	**NDCG@5**	**NDCG@10**	**Hit@1**	**Hit@5**	**Hit@10**	**NDCG@5**	**NDCG@10**
Any Collab.	Majority	0.07	0.27	0.50	0.17	0.24	0.10	0.37	0.69	0.23	0.33
	Majority (author)	11.4	38.3	60.7	24.7	31.9	10.2	30.6	50.1	20.4	26.7
	Graph	4.5	15.4	22.7	10.1	12.4	5.2	17.7	25.5	11.6	14.1
	BERT4Rec	39.3	57.6	65.4	49.0	51.5	22.5	50.0	65.0	36.7	41.6
	CoBERT	**71.7**	**81.4**	**84.8**	**76.9**	**78.0**	**78.1**	**88.0**	**91.1**	**83.5**	**84.5**
New Collab.	Majority	0.05	0.17	0.29	0.11	0.15	0.07	0.21	0.38	0.14	0.20
	Majority (author)	—	—	—	—	—	—	—	—	—	—
	Graph	8.7	16.1	22.1	12.5	14.4	11.4	17.7	22.8	14.6	16.2
	BERT4Rec	16.4	35.1	45.1	26.2	29.4	27.3	47.8	59.9	37.9	41.8
	CoBERT	**58.4**	**68.9**	**72.9**	**64.1**	**65.4**	**69.1**	**79.5**	**83.1**	**74.8**	**75.9**

TABLE III: Results on the **PlosOne** dataset. Given are the evaluation metrics' means over five runs. All values are in percent. Best values are written in bold. The majority baseline per author cannot be evaluated for the New Collaborator Prediction task, since it does not predict any new collaborators.

		$\mathbf{n = 5}$					$\mathbf{n = 10}$				
Task	**Model**	**Hit@1**	**Hit@5**	**Hit@10**	**NDCG@5**	**NDCG@10**	**Hit@1**	**Hit@5**	**Hit@10**	**NDCG@5**	**NDCG@10**
Any Collab.	Majority	0.02	0.10	0.20	0.06	0.09	0.02	0.13	0.26	0.08	0.12
	Majority (author)	5.4	23.3	**47.2**	14.0	21.7	5.0	21.2	41.7	12.9	19.5
	Graph	10.8	34.1	43.0	22.9	25.8	7.7	28.9	39.7	18.4	21.9
	BERT4Rec	9.9	26.3	37.1	18.3	21.8	34.8	56.9	66.2	46.5	49.5
	CoBERT	**30.5**	**39.3**	43.7	**35.1**	**36.6**	**48.2**	**58.8**	**63.2**	**53.9**	**55.3**
New Collab.	Majority	0.01	0.02	0.03	0.02	0.02	0.00	0.04	0.05	0.02	0.02
	Majority (author)	—	—	—	—	—	—	—	—	—	—
	Graph	6.9	13.5	19.5	10.3	12.2	7.7	13.4	19.1	10.6	12.4
	BERT4Rec	8.8	24.6	35.9	16.8	20.4	23.3	44.6	55.8	34.4	38.0
	CoBERT	**28.3**	**38.0**	**42.8**	**33.4**	**35.0**	**46.4**	**58.2**	**63.1**	**52.7**	**54.3**

TABLE IV: Ablation study based on the AI dataset with $n = 5$ in the predicting any collaborator task.

	Hit@k			**NCDG@k**	
Model	$\mathbf{k = 1}$	$\mathbf{k = 5}$	$\mathbf{k = 10}$	$\mathbf{k = 5}$	$\mathbf{k = 10}$
CoBERT	71.68	81.36	84.81	76.92	78.03
w/o ACE	70.87	81.23	84.76	76.48	77.63
w/o PCE	61.70	76.59	82.00	69.71	71.46
w/o ACE / PCE	67.45	79.21	83.53	73.79	75.19
w/o ACE / PCE / PPE (= BERT4Rec)	39.25	57.55	65.40	48.97	51.51

BERT4Rec model that make up CoBERT. Namely, these are the positional encoding per paper (PPE), the paper content embedding (PCE), and the co-author content embedding (ACE). We thus train CoBERT models without adding the corresponding embeddings in the embedding layer or, in case of ACE, learning an embedding per author from scratch. Results are shown in Tables IV and V for the task of predicting any and new collaborators, respectively, with $n = 5$ on the AI dataset. Again, the mean results over five runs are given. The results for the other task, minimal sequence length, and dataset show similar trends and can be found in Appendix A.

The ablation study shows that our proposed modifications mostly result in better model performance. Removing the paper content embedding (*w/o PCE*) induces a substantial drop in performance, showing that content based features are important for the model. On the other hand, the prediction quality varies greatly when removing the co-author content embedding (*w/o ACE*). While sometimes, the performance drops, the model sometimes also improves. We suspect that the model already learns some form of co-author content representation implicitly since it sees paper content embeddings per co-author during training. Thus an explicit co-author content embedding is not necessarily important. Overall, the performance gain in most of our experiments was larger than the usually less pronounced reduction.

Removing all three proposed modifications (*w/o ACE / PCE / PPE*) makes CoBERT equivalent to the BERT4Rec model, which scores low results. BERT4Rec performs much better when adding PPE (*w/o ACE / PCE*). For BERT4Rec, each co-author gets an unique positional encoding, thus making the order of co-authors for one paper arbitrary. Using PPE, we assume that the order of co-authors for one paper is not important. Thus, the model can learn to find patterns in the sequence regarding the paper order.

V. Conclusion

In this paper, we have proposed CoBERT, a BERT based model to recommend scientific collaborations, by interpreting

TABLE V: Ablation study based on the AI dataset with $n = 5$ in the predicting new collaborator task.

	Hit@k			NCDG@k	
Model	$\mathbf{k = 1}$	$\mathbf{k = 5}$	$\mathbf{k = 10}$	$\mathbf{k = 5}$	$\mathbf{k = 10}$
CoBERT	58.42	68.93	72.94	64.09	65.39
w/o ACE	64.44	74.97	78.60	70.18	71.35
w/o PCE	38.66	53.70	60.09	46.68	48.75
w/o ACE / PCE	47.58	60.45	65.69	54.47	56.17
w/o ACE / PCE / PPE (= BERT4Rec)	16.45	35.13	45.06	26.17	29.37

the setting as a sequential recommendation task. For this, we introduced multiple changes to the sequential recommendation model BERT4Rec that improve the performance of the model on the task of predicting the next co-author in an author's co-author sequence. The performance gain was steady regarding common baselines and over multiple datasets and tasks.

Common scientific collaboration recommendation models rely on collaboration graphs, which need to be kept in memory to efficiently compute embeddings. Since CoBERT does not rely on graphs, all components, i.e. the co-author content and paper content embeddings, can be computed and processed independently. Thus, CoBERT can potentially be applied to larger datasets than graph based methods. In the future, we want to verify this by applying CoBERT to settings with multi millions of papers and authors, e.g. by using the full Semantic Scholar dataset.

In our ablation study, we found that adding co-author content leads to unsteady model performance. Future work might want to investigate this further and search for better initialization for this embedding.

Finally, since we insert features regarding the paper contents, we are also able to make thematically fitting co-author predictions. Instead of masking the co-author content embedding and paper content embeddings for the collaboration recommendation task, we could only mask the co-author content embedding and use a proposal abstract as the paper embedding. In this way, the model would take paper content into account when recommending co-authors. We will explore this and other potential use cases for CoBERT in the future.

REFERENCES

[1] R. Boschma, "Proximity and innovation: A critical assessment," *Regional Studies*, vol. 39, no. 1, 2005.

[2] A. Grover and J. Leskovec, "Node2Vec: Scalable feature learning for networks," in *SIGKDD*, 2016.

[3] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *WWW*, 2015.

[4] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using siamese BERT-networks," *arXiv preprint arXiv:1908.10084*, 2019.

[5] S. Wang, L. Hu, Y. Wang, L. Cao, Q. Z. Sheng, and M. Orgun, "Sequential recommender systems: Challenges, progress and prospects," in *IJCAI*, 2019.

[6] E. Fischer, D. Zoller, A. Dallmann, and A. Hotho, "Integrating keywords into BERT4Rec for sequential recommendation," in *KI 2020: Advances in Artificial Intelligence*, 2020.

[7] R. Guns and R. Rousseau, "Recommending research collaborations using link prediction and random forest classifiers," *Scientometrics*, vol. 101, no. 2, 2014.

[8] Y. Sun, R. Barber, M. Gupta, C. C. Aggarwal, and J. Han, "Co-author relationship prediction in heterogeneous bibliographic networks," in *ASONAM*, 2011.

[9] T. Koopmann, M. Stubbemann, M. Kapa, M. Paris, G. Buenstorf, T. Hanika, A. Hotho, R. Jäschke, and G. Stumme, "Proximity dimensions and the emergence of collaboration: A HypTrails study on German AI research," *Scientometrics*, 2021.

[10] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *SIGKDD*, 2014.

[11] T. K. T. Ho, Q. V. Bui, and M. Bui, "Co-author relationship prediction in bibliographic network: A new approach using geographic factor and latent topic information," in *Proceedings of the Tenth International Symposium on Information and Communication Technology*, 2019.

[12] Q. Zhang and H. Yu, "Computational approaches for predicting biomedical research collaborations," *PLOS ONE*, vol. 9, no. 11, 2014.

[13] A. Daud, M. Ahmad, M. S. I. Malik, and D. Che, "Using machine learning techniques for rising star prediction in co-author network," *Scientometrics*, vol. 102, no. 2, 2015.

[14] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang, "BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer," in *CIKM*, 2019.

[15] W.-C. Kang and J. McAuley, "Self-attentive sequential recommendation," in *ICDM*, 2018.

[16] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," *arXiv preprint arXiv:1511.06939*, 2015.

[17] J. Tang and K. Wang, "Personalized top-n sequential recommendation via convolutional sequence embedding," in *WSDM*, 2018.

[18] C. Xu, P. Zhao, Y. Liu, J. Xu, V. S. S.Sheng, Z. Cui, X. Zhou, and H. Xiong, "Recurrent convolutional neural network for sequential recommendation," in *WWW*, 2019.

[19] R. Ren, Z. Liu, Y. Li, W. X. Zhao, H. Wang, B. Ding, and J.-R. Wen, "Sequential recommendation with self-attentive multi-adversarial network," in *SIGIR*, 2020.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS*, vol. 30, 2017.

[21] T. X. Tuan and T. M. Phuong, "3D convolutional networks for session-based recommendation with content features," in *RecSys*, 2017.

[22] B. Hidasi, M. Quadrana, A. Karatzoglou, and D. Tikk, "Parallel recurrent neural network architectures for feature-rich session-based recommendations," in *RecSys*, 2016.
[23] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *NeurIPS*, 2019.
[24] K. Kersting, J. Peters, and C. Rothkopf, "Was ist eine Professur für Künstliche Intelligenz?" *arXiv preprint arXiv:1903.09516*, 2019.
[25] W. Ammar, D. Groeneveld, C. Bhagavatula, I. Beltagy, M. Crawford, D. Downey, J. Dunkelberger, A. Elgohary, S. Feldman, V. Ha, *et al.*, "Construction of the literature graph in Semantic Scholar," *arXiv preprint arXiv:1805.02262*, 2018.
[26] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in *SIGKDD*, 2014.
[27] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv preprint arXiv:1412.6980*, 2017.
[28] L. Prechelt, "Early Stopping - But When?" In *Neural Networks: Tricks of the Trade*, vol. 1524, 1998.
[29] L. H. Li, M. Yatskar, D. Yin, C.-J. Hsieh, and K.-W. Chang, "What does BERT with vision look at?" In *ACL*, 2020.

APPENDIX

Here, we show additional results for our experiments, including model adaptions that unfortunately did not lead to any improvement.

A. Remaining Results of the Ablation Study

Tables VIII to XIII show the ablation studies for the remaining datasets and task variations. As already shown, the benefit of using co-author content embeddings (ACE) varies greatly.

B. Things That Didn't Work

In addition to the final CoBERT approach, we tested two additional variations of the sequential approach for collaboration recommendation: (1) Calculating a weighted sum of the proposed embeddings and jointly learning their weights with the model parameters and (2) separating authors from papers in the input sequence. Both approaches did not work well. Results compared to CoBERT are shown in Tables VI and VII.

1) Weighted Embeddings: The input representation for the transformer layer is generated in the embedding layer (c.f. Figure 2). While CoBERT adds all the embeddings to obtain one initial representation per item, we experiment with trainable weights λ_i that are optimized jointly with the network parameters:

$$\mathbf{h}_i^0 = \underbrace{\lambda_{\text{PPE}} \cdot \mathbf{p}_i}_{\substack{\text{Positional Encoding} \\ \text{per Paper}}} + \underbrace{\lambda_{\text{PCE}} \cdot \mathbf{c}_i}_{\substack{\text{Paper Content} \\ \text{Embedding}}} + \underbrace{\lambda_{\text{ACE}} \cdot \mathbf{a}_i}_{\substack{\text{Co-Author Content} \\ \text{Embedding}}} . \quad (3)$$

We initialize all $\lambda_i = 1$ and call this model **CoBERT-Weighted (CoBERT-W)**. The performance for this approach drops significantly compared to CoBERT. Inspecting the trained embedding weights for the PlosOne models show that the network mostly relies on the co-author content embedding (ACE): λ_{PPE} and λ_{PCE} are set around zero. For the AI dataset, the paper content embedding (PCE) is also weighted similarly as the ACE. The positional encoding is not considered by CoBERT-W, which might be one of the reasons why this method does not perform well.

Possible ways to improve this approach are to exclude weight learning for the positional encoding or normalizing the weights using a softmax function, restricting the scaling of the model input.

2) Sequential Approach: VisualBERT [29] has introduced an approach, which appends two sequences separated by a `[SEP]` token [29]. We adopt this approach and split the input sequence into co-authors and papers. Here, the co-authors of a paper in the first sequence and the respective paper in the second sequence have the same positional embeddings. Both sequences are separated using the `[SEP]` token. The model is supposed to learn the weighting and correlation of papers to authors inherently. We call this model **CoBERT-Seq**.

Due to the separation, the input sequence gets longer, which can, in exceptional cases, result in sequences that are trimmed to the model's maximal number of input tokens. Also, we find that this approach does not improve the model's performance compared to CoBERT. While it performs relatively well on the AI dataset compared to the default BERT4Rec model or CoBERT-W, it shows very low scores on the PlosOne dataset.

TABLE VI: Results of our other CoBERT variants on the **AI** dataset. Given are the evaluation metrics' means over five runs. All values are in percent. Best values are written in bold.

		$\mathbf{n=5}$					$\mathbf{n=10}$				
Task	**Model**	**Hit@1**	**Hit@10**	**Hit@5**	**NDCG@5**	**NDCG@10**	**Hit@1**	**Hit@5**	**Hit@10**	**NDCG@5**	**NDCG@10**
Any Collab.	CoBERT	**71.7**	**81.4**	**84.8**	**76.9**	**78.0**	**78.1**	**88.0**	**91.1**	**83.5**	**84.5**
	CoBERT-W	27.8	40.4	46.5	34.4	36.4	47.7	63.7	70.0	56.2	58.3
	CoBERT-Seq	46.7	60.3	65.9	53.9	55.7	69.5	84.3	88.6	77.5	78.9
New Collab.	CoBERT	**58.4**	**68.9**	**72.9**	**64.1**	**65.4**	**69.1**	**79.5**	**83.1**	**74.8**	**75.9**
	CoBERT-W	18.0	30.2	36.9	24.4	26.5	33.0	50.2	57.8	42.0	44.5
	CoBERT-Seq	41.5	57.6	64.3	50.1	52.3	55.1	71.8	77.7	64.1	66.0

TABLE VII: Results of our other CoBERT variants on the **PlosOne** dataset. Given are the evaluation metrics' means over five runs. All values are in percent. Best values are written in bold.

		$\mathbf{n=5}$					$\mathbf{n=10}$				
Task	**Model**	**Hit@1**	**Hit@10**	**Hit@5**	**NDCG@5**	**NDCG@10**	**Hit@1**	**Hit@5**	**Hit@10**	**NDCG@5**	**NDCG@10**
Any Collab.	CoBERT	**30.5**	**39.3**	**43.7**	**35.1**	**36.6**	**48.2**	**58.8**	**63.2**	**53.9**	**55.3**
	CoBERT-W	1.1	4.8	9.2	3.0	4.3	13.6	23.0	29.3	18.5	20.5
	CoBERT-Seq	1.0	4.5	8.8	2.7	4.1	3.7	9.5	14.9	6.6	8.3
New Collab.	CoBERT	**28.3**	**38.0**	**42.8**	**33.4**	**35.0**	**46.4**	**58.2**	**63.1**	**52.7**	**54.3**
	CoBERT-W	11.6	20.8	27.3	16.3	18.4	9.1	16.4	21.8	12.8	14.6
	CoBERT-Seq	0.9	4.1	8.0	2.4	3.7	18.3	27.5	33.3	23.0	24.9

TABLE VIII: Ablation study based on the AI dataset with $n = 10$ in the predicting any collaborator task.

		Hit@k		**NCDG@k**	
Model	**Hit@1**	$\mathbf{k=5}$	$\mathbf{k=10}$	$\mathbf{k=5}$	$\mathbf{k=10}$
CoBERT	78.11	88.01	91.13	83.52	84.53
w/o ACE	78.88	88.13	90.97	83.92	84.84
w/o PCE	66.25	81.48	86.53	74.48	76.12
w/o ACE / PCE	71.21	82.88	86.88	77.54	78.84
w/o ACE / PCE / PPE (= BERT4Rec)	22.55	50.03	65.00	36.73	41.57

TABLE IX: Ablation study based on the AI dataset with $n = 10$ in the predicting new collaborator task.

		Hit@k		**NCDG@k**	
Model	**Hit@1**	$\mathbf{k=5}$	$\mathbf{k=10}$	$\mathbf{k=5}$	$\mathbf{k=10}$
CoBERT	69.08	79.50	83.14	74.77	75.95
w/o ACE	65.32	76.36	80.21	71.31	72.55
w/o PCE	58.32	76.18	82.09	68.01	69.93
w/o ACE / PCE	59.26	73.62	78.72	67.03	68.69
w/o ACE / PCE / PPE (= BERT4Rec)	27.30	47.77	59.92	37.91	41.83

TABLE X: Ablation study based on the PlosOne dataset with $n = 5$ in the predicting any collaborator task.

		Hit@k		**NCDG@k**	
Model	**Hit@1**	$\mathbf{k=5}$	$\mathbf{k=10}$	$\mathbf{k=5}$	$\mathbf{k=10}$
CoBERT	30.53	39.26	43.71	35.14	36.57
w/o ACE	15.92	22.13	26.53	19.13	20.54
w/o PCE	13.51	22.17	28.06	17.98	19.87
w/o ACE / PCE	9.53	17.04	22.62	13.37	15.15
w/o ACE / PCE / PPE (= BERT4Rec)	4.45	12.96	19.74	8.75	10.92

TABLE XI: Ablation study based on the PlosOne dataset with $n = 5$ in the predicting new collaborator task.

		Hit@k		**NCDG@k**	
Model	**Hit@1**	$\mathbf{k=5}$	$\mathbf{k=10}$	$\mathbf{k=5}$	$\mathbf{k=10}$
CoBERT	28.33	37.98	42.81	33.42	34.98
w/o ACE	29.03	38.89	43.98	34.22	35.86
w/o PCE	26.18	36.66	42.53	31.66	33.55
w/o ACE / PCE	22.10	31.39	36.64	26.96	28.65
w/o ACE / PCE / PPE (= BERT4Rec)	8.77	24.57	35.86	16.78	20.42

TABLE XII: Ablation study based on the PlosOne dataset with $n = 10$ in the predicting any collaborator task.

		Hit@k		**NCDG@k**	
Model	**Hit@1**	$\mathbf{k=5}$	$\mathbf{k=10}$	$\mathbf{k=5}$	$\mathbf{k=10}$
CoBERT	48.20	58.84	63.18	53.90	55.31
w/o ACE	46.22	57.51	62.40	52.23	53.81
w/o PCE	19.68	32.96	40.86	26.56	29.11
w/o ACE / PCE	12.60	22.89	29.98	17.88	20.16
w/o ACE / PCE / PPE (= BERT4Rec)	34.77	56.89	66.23	46.52	49.54

TABLE XIII: Ablation study based on the PlosOne dataset with $n = 10$ in the predicting new collaborator task.

		Hit@k		**NCDG@k**	
Model	**Hit@1**	$\mathbf{k=5}$	$\mathbf{k=10}$	$\mathbf{k=5}$	$\mathbf{k=10}$
CoBERT	46.37	58.20	63.14	52.66	54.26
w/o ACE	46.46	57.84	62.58	52.53	54.06
w/o PCE	44.92	56.64	62.29	51.11	52.94
w/o ACE / PCE	38.66	49.99	55.51	44.68	46.46
w/o ACE / PCE / PPE (= BERT4Rec)	23.32	44.58	55.78	34.38	37.99