COMBINATORIAL SCIENTIFIC DISCOVERY: FINDING NEW CONCEPT COMBINATIONS BEYOND LINK PREDICTION

Yuchen Zeng, Yelman Khan , Shufan Wang
College of Information and Computer Sciences
University of Massachusetts Amherst
{yuchenzeng,yelmankhan,shufanwang}@cs.umass.edu

Veronika Thost* & Tengfei Ma*
MIT-IBM Watson AI Lab, IBM Research
vth@zurich.ibm.com, tengfei.ma1@ibm.com

ABSTRACT

As the number of publications is growing tremendously, it is more and more a challenge for researchers to read all related literature to find the ”white space” in a specific research domain. Automatic scientific discovery has been proposed to help researchers identify new research ideas, but it has generally been limited to finding new combinations of concept pairs using link prediction in a knowledge graph. In this paper, we propose the combinatorial scientific discovery task: predicting combinations of more than two concepts. We standardize the task by providing benchmark datasets and initial models. Our solutions demonstrate the challenge but also the value of the task to find new, meaningful scientific ideas and its advantage over simple link prediction.

1 INTRODUCTION

In the past years, the number of new publications has grown tremendously, especially in AI. This leads to many challenges for researchers: (I) What papers outside my research area should I read? (II) What research questions should I investigate based on the recent literature? (III) What new solutions can I use for potential problems? All these questions are related to the *scientific discovery problem*. In this paper, we define scientific discovery as finding new research ideas in some scientific domain. We approach this problem in terms of finding new combinations of scientific concepts worth to study.

Despite of the importance of the scientific discovery problem has been recognized recently,[1] the studies on this problem are rather limited. Previous works mostly focus on predicting unknown links between two concept nodes in a constructed knowledge graph. For example, in the biomedical domain, Luo et al. (2018) extracted a graph of concept nodes (e.g., “chronic infection”, “malnutrition”) connected by rather domain-specific so-called influence links (e.g., “causes”, “activates”) from the Pubmed corpus, added several hand-crafted features (e.g., degree, similarity scores) to the nodes, and then did link prediction using basic classifiers (e.g., Random Forests). Further, the small size of the data (˜5k positive samples) makes it less useful for deep learning. Wang et al. (2019) also generated new ideas based on link prediction and went further generating paper drafts.

However, real scientific discovery goes beyond link prediction. In many cases, the combination of only two concepts cannot capture the exact meaning of a new idea (see Figure 1). The search for new concept combinations should not be restricted to pairs of concepts, and we claim that new models and evaluation scenarios are needed to generate more realistic ideas that are high-order

*Corresponding Author

[1] https://www.iarai.ac.at/science4cast/

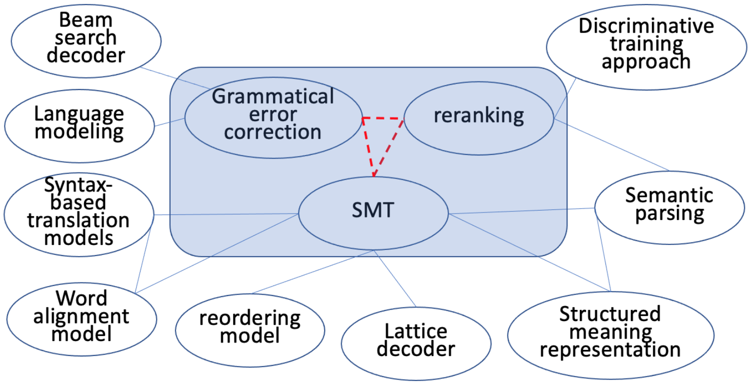
Beam search decoder
Discriminative training approach
Language modeling
Grammatical error correction
reranking
Syntax-based translation models
SMT
Semantic parsing
Word alignment model
reordering model
Lattice decoder
Structured meaning representation

Figure 1: Example of combinatorial scientific discovery. A new idea (NAACL paper "Discriminative Reranking for Grammatical Error Correction with Statistical Machine Translation") consisting of three concepts. A single link between any two of them does not fully capture the idea, thus a standard link predictor detecting only pairwise combinations cannot be applied to predict this new research idea.

combinations. We call this new task "*combinatorial scientific discovery*". In this paper, we provide a basis for studying combinatorial scientific discovery in the AI community.[2]

- We created two datasets, **CSD-NLP** & **CSD-CV**, built from papers in NLP and CV conferences, which allow to study the **C**ombinatorial **S**cientific **D**iscovery problem (see Table 1 for an overview). For each dataset, we extracted the concepts in these corpora and constructed a scientific knowledge graph. We extracted the facts about high-order concept combinations in the graph and then split the data for training, validation and testing. Together with the data, we also make available all necessary scripts so that similar datasets can be build in only minutes.
- We use a pre-trained BERT language model and graph neural networks to create concept embeddings and explore different solutions based on those, which may serve as baselines: one standard link prediction model and four combinatorial prediction models, predicting concept combinations beyond pairs, including a prompt based method and a new energy-like scoring function.
- Our experiments show that standard link predictors cannot directly be applied for predicting concept combinations beyond pairs successfully. Our combination predictors shows clear advantages and lay a foundation for studying the challenging combinatorial scientific discovery task.

2 Related Work

Automatic scientific discovery has only been considered in few previous works Luo et al. (2018); Wang et al. (2019); Krenn & Zeilinger (2020). Yet, as outlined in Section 1, the data model in Luo et al. (2018) is based on domain-specific information (i.e., influence links); the dataset is rather small (˜5k positive samples) and focuses only on biomedical domains; the models use basic ML based on hand-engineered features and, in particular, only do link prediction. Wang et al. (2019) propose PaperRobot which, given a paper title, predicts related entities and generates some key elements of a new paper. However, it also focuses on medical papers and determine new concepts using link prediction in a very diverse graph. Without well-defined relationships in the graph and enough data, their case study in the NLP domain shows much less satisfactory results. Krenn & Zeilinger (2020) built semantic networks in quantum physics and predict the research trends by link prediction. Our dataset and models solve these issues and hence provide a good start and generalization for deep learning research on this problem.

[2] https://github.com/matenure/sci-disco

	CSD-NLP	**CSD-CV**
# papers considered	7,605	19,877
# concept nodes	9,819	36,172
# correlation links	18,227	65,139
average node degree	4.47	3.43
graph diameter	11	14
# train/valid/test papers	5,641/774/484	13,466/2,055/2,969
# train/valid/test papers w/ concepts S, $\|S\| > 2$	4,141/578/384	10,937/1,713/2,579

Table 1: Statistics about our datasets. The last line shows the importance of concept combinations beyond pairs.

	CSD-NLP				CSD-CV			
Model	F1	Hits@10	Hits@20	Hits@30	F1	Hits@10	Hits@20	Hits@30
LP	29.08	2.53	6.21	14.07	15.92	2.08	4.81	10.78
CP-pro	37.44	**6.53**	14.88	**25.23**	53.89	**14.38**	**30.12**	**44.84**
CP-avg	49.19	1.30	5.98	16.57	47.25	2.25	10.47	25.41
CP-gnn	48.39	5.87	**15.20**	23.45	41.64	4.63	12.93	24.20
CP-enb	**61.61**	4.16	12.60	25.03	**64.59**	10.21	26.97	44.42

Table 2: Overview of results.

Other works are only coarsely related to scientific discovery, but they focus on very different topics, such as citation field extraction Thai et al. (2020), citation analysis Mohammad (2020), multi-document summarization of scientific articles Lu et al. (2020), or the transfer of scientific concepts across text corpora Cao et al. (2020).

3 The CSD Datasets

Table 1 gives an overview of our datasets. We consider papers from three major NLP and three CV conferences, details about the paper collection and curation are given in Appendix A. Note that the sizes of the datasets are very different, with the CV data being much larger, thus yielding very different test scenarios. The extraction of the concept nodes, which is more involved, is detailed in a subsection below. We create an undirected link between two nodes when the corresponding concepts occur together in one paper; recall that we restrict the focus to the abstracts for mitigating noise. We created time-based train/valid/test splits based on (concept combinations co-occurring in) papers and modeling a realistic prediction scenario, as suggested in Hu et al. (2020); the train/valid/test set contain all papers from 2000-2014/2015/2016. Links between concepts that co-occur in a paper in the valid or test set are removed to prevent information leakage.

3.1 Extraction of Concept Nodes

We used the pretrained SciERC model Wadden et al. (2019) to perform entity extraction on the collected abstracts. The model was pretrained on a dataset including annotations for scientific entities for 500 scientific paper abstracts from major AI conferences. Unlike previous works such as Wang et al. (2019), we did not extract all entities and relations but just focused on entities of predicted type "method". By focusing on combining only method concepts, i.e., the main components of a scientific idea, we also avoid the uncertainty of relationship prediction. Since the entity extraction results include multiple lexical forms of the same concept, we clustered them to create our final concepts. We calculated the similarities of all candidates based on their surface forms, and merge similar ones. More details can be found in Appendix A.

	CSD-NLP Test Set			
Model	F1	Hits@10	Hits@20	Hits@30
CP-enb	61.10	4.64	14.89	26.78

Table 3: Results for combined NLP and CV train data.

NLP	word alignment models, SMT, spectral embedding, unsupervised language models
	look-ahead methods, discriminative classifier, pruning framework
	probability integral transform, Moses, phrase-based system, time-based merging
	structured perceptron, data-driven sense induction method, parser, joint learning
NLP+CV	language models, discriminating models, joint model, refined multi-scale image segmentations
	image ranker learning, head-shoulder detector, unsupervised syntactic parser, multiview stereo
	cubic splines, bag-of-words, domain adaptation, MDL and SPHARM methods

Table 4: Concept combinations CP-enb scores high when learning over CSD-NLP (top) and both NLP and CV.

.

3.2 Evaluation of Concept Qualities

To validate that the extracted concepts are good enough for our task, we evaluate our data in terms of two perspectives: the quality of entity extraction, the quality of entity clustering. Specifically, we randomly selected 3902 papers in the dataset, and manually filtered unreasonable entities and combined similar entities. Within the 16227 original extracted entities, 11794 entities remained after manually filtering, which means about 80.5% entities are valid in our original extractions. Moreover, we compare the clustered results with the manual annotations, the precision is 90.9%, recall is 73.2%, and f1 is 81.1%. After automatically clustering, we have less entities than the manual filtering, but the precision score is high. These results indicate that most of the remaining entities after entity extraction and clustering are valid, i.e. they can be regarded as good concepts by humans. We also measure other statistics of the construct4d concept graph in Table 1. The average node degree is 4.47 and 3.43 for the two datasets separately, which indicates that the graph is not too sparse and we have enough connections for training; and most of the papers contain more than 2 concepts.

4 Evaluation

We conducted initial experiments over our CSD datasets in order to point out the challenges of combinatorial scientific discovery. We explore different approaches as initial solutions for this problem, which may serve as baselines in future works.

4.1 Models

Given a set S of concepts, our models aim to predict whether the combination of these concepts in S is new and realistic. We compare a standard **link prediction model (LP)**, which assumes $|S| = 2$, with several **higher-order combination prediction models (CP)**, which are able to predict combinations of more than two concepts.

LP: The link prediction model (1) uses BERT to convert concept nodes into 768-dimensional word embeddings as node features; (2) applies a graph convolutional network (GCN) Kipf & Welling (2017) to obtain concept node embeddings $\{\mathbf{c}_1, \mathbf{c}_2\}$ from the 768-dimensional word embeddings for the concepts in S; (3) combines these embeddings: $\mathbf{c}_{\mathbf{LP}}(S) = \mathbf{c}_1 * \mathbf{c}_2$; and (4) uses an MLP (multi-layer perception) to predict the label $\mathbf{LP}(S)$.

As most test samples contain more than two concepts, in order to evaluate the link prediction model, we need to extend the link prediction results to combinations S with $|S| > 2$. In this context, we make a simple assumption: a combination is true only if all pairwise links in the set are true:

$$\text{LP}(S) = \begin{cases} 1 & \text{if LP}((c_i, c_j)) > 0.5 \\ & \text{for all } i, j \in |S|, i < j \\ 0 & \text{otherwise.} \end{cases}$$

Leveraging the pre-trained language model BERT Devlin et al. (2019), prompted-based learning Brown et al. (2020), and energy-like score functionsBishop (2006), we propose four different higher-order combination prediction models (CP). In all these model, we use a scoring function to first get the score for a combination S and then predict the probability $p(S) = \text{Sigmoid}(\text{Score}(S))$.

- **CP-pro.** Prompted-based learning: For each combination S, we consider all pairwise subset combinations $\{C_1, C_2\} \subseteq S$, and combine them with a prefix (hard prompt) of form "This paper uses C_1 and C_2." We use BERT to get the representation of the sentences and then use MLP layers to obtain the final scores.
- **CP-avg.** Simple average scoring function: Instead of directly encoding the entire combination with BERT, we encode each concept in S with BERT and then average these embeddings to represent S. Then, we use the same MLP as above to predict the final score.
- **CP-gnn.** Extension of CP-avg: after BERT, we apply a GCN to obtain deeper embeddings.
- **CP-enb.** We design an energy-based scoring function replacing the average in CP-avg, to encode both atom and pair-wise information:

  $$Score(\mathbf{c}_1, ..., \mathbf{c}_{|S|}) = \tfrac{1}{|S|} \textstyle\sum_i \mathbf{c}_i^T \mathbf{b}_i + \tfrac{1}{N} \textstyle\sum_{i<j} (\mathbf{c}_i^T \mathbf{W} \mathbf{c}_j),$$

  c_i is the embedding of a concept in S, N is the number of all pairwise combinations in S.

Further details about the model configurations and training can be found in Appendix B.

4.2 Results

An overview of our results is given in Table 2. All numbers are averaged over three runs with different random seeds. The disappointing results of LP clearly show that it is not a good option to simply combine the link prediction scores for a larger combination. Standard link predictors cannot be easily extended for predicting combinations of more than two concepts. In contrast, the CP-enb model which uses our higher-order scoring function achieves the overall best results in terms of F1. While the CP-pro (BERT prompt) achieves highest Hits@K, its performance on F1 is much worse, and its advantage over other models on Hits@20 and Hits@30 is also rather small. That indicates that the prompt model can predict the highest ranked matchings better, but its overall performance over all combinations is not as good as the energy function based model CP-enb. Comparing CP-avg and CP-enb, we can clearly see the scoring function used in CP-enb is superior to simple averaging. Finally, to our surprise, CP-gnn does not perform well. Possible reasons may be that the model is too complex and the graph is too sparse (i.e., not enough train data).

We also tested the cross-domain effects. Table 3 shows the results for the models trained over an augmented NLP dataset by combining CSD-NLP and CSD-CV. More precisely, we add the CV conference papers during 2000-2014 to the existing CSD-NLP training dataset for joint training, and then test on the same test data as CSD-NLP (shown in Table 2). We see that the additional domain knowledge improves the scores with the NLP part with respect of Hits@k. This shows that adding the knowledge even from other domains may also bring benefits to the prediction of new ideas. Interestingly, we did not see major improvements when performing this experiment for the larger CV data.

4.3 Example Predictions

Table 4 shows examples of new combinations obtaining high probabilities in our model and not occurring in our paper corpus. Many of these look surprisingly reasonable. We also see that, when we combine the NLP and CV papers during training, we might learn even more interesting combinations. Note that our current data does not include the most recent years, which limits the "imaginative power" of the system. In the future, we plan to collect more recent papers to generate more "modern" ideas.

5 Conclusions

In this paper, we study the scientific discovery problem which, in particular, has become more severe with the fast-growing number of publications in AI. We propose the new task as predicting combinations of more than two concepts. We provide an initial framework to create corresponding datasets, together with solid initial solutions. Although there are still some limitations (e.g., the extracted concepts have some noise), we believe our motivation, datasets, and solutions will inspire future works.

6 Acknowledgements

This project is part of the data science industry mentorship program initiated by Andrew McCallum at University of Massachusetts Amherst. We thank the teaching assistants Xiang Lorraine Li and Rico Angell for helpful discussion and the instructor Andrew McCallum for valuable feedback. Experiments in this project are conducted on the Gypsum cluster at UMassAmherst. The cluster is purchased with funds from the Massachusetts Technology Collaborative.

References

Christopher M Bishop. Pattern recognition and machine learning. 2006.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *ArXiv*, abs/2005.14165, 2020.

Hancheng Cao, Mengjie Cheng, Zhepeng Cen, Daniel A. McFarland, and Xiang Ren. Will this idea spread beyond academia? understanding knowledge transfer of scientific concepts across text corpora. In Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proc. of EMNLP*, pp. 1746–1757. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.findings-emnlp.158.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805, 2019.

Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In *Proc. of NeurIPS*, 2020.

Thomas Kipf and Max Welling. Semi-supervised learning with graph convolutional neural networks. In *Proc. of ICLR*, 2017.

Mario Krenn and Anton Zeilinger. Predicting research trends with semantic and neural networks with an application in quantum physics. *Proceedings of the National Academy of Sciences*, 117 (4):1910–1916, 2020.

Yao Lu, Yue Dong, and Laurent Charlin. Multi-XScience: A large-scale dataset for extreme multi-document summarization of scientific articles. In *Proc. of EMNLP*, pp. 8068–8074. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.648.

Fan Luo, Marco A. Valenzuela-Escárcega, Gus Hahn-Powell, and Mihai Surdeanu. Scientific discovery as link prediction in influence and citation graphs. In *Proc. of TextGraphs-12*, pp. 1–6. Association for Computational Linguistics, 2018. doi: 10.18653/v1/W18-1701.

Saif M. Mohammad. Examining citations of natural language processing literature. In *Proc. of ACL*, pp. 5199–5209. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.464.

Conference	# Papers	# Papers w/ abstract
EMNLP	1,845	1,695
ACL	3,521	3,138
NAACL	2,239	2,066
CVPR	12,317	12,266
ICCV	4,706	4,052
ECCV	2,854	2,176

Table 5: Overview of conferences considered for paper extraction for CSD-NLP (top) and CSD-CV (bottom). We only used papers which were accompanied by an abstract.

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

Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. Arnetminer: extraction and mining of academic social networks. In Ying Li, Bing Liu, and Sunita Sarawagi (eds.), *Proc. of SIGKDD*, pp. 990–998. ACM, 2008. doi: 10.1145/1401890.1402008.

Dung Thai, Zhiyang Xu, Nicholas Monath, Boris Veytsman, and Andrew McCallum. Using bibtex to automatically generate labeled data for citation field extraction. In *Proc. of AKBC*, 2020. doi: 10.24432/C5F592.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. Entity, relation, and event extraction with contextualized span representations. In *Proc. of EMNLP-IJCNLP*, pp. 5784–5789, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1585.

Qingyun Wang, Lifu Huang, Zhiying Jiang, Kevin Knight, Heng Ji, Mohit Bansal, and Yi Luan. Paperrobot: Incremental draft generation of scientific ideas. In *Proc. of ACL*, pp. 1980–1991, 2019.

A Additional Details about CSD

We used the papers provided in the DBLP-Citation-network V10[3] for our project Tang et al. (2008). An overview of the conferences considered for CSD, including paper numbers, is given in Table 5. Specifically, we considered only papers from the years 2000-2016 w/ abstract information.

For concept clustering, we represent each candidate concept as a set of n-grams, and weight them with TF-IDF. Then we compute the cosine similarities of any two concepts. If their similarity is larger than our threshold, we merge them together and use the longer one to represent the concept. For example, if we have two extracted candidate concepts, ”cluster” and ”clustering”, we have the following procedures to merge them: (1) n-grams representations: ”cluster”=[”clu”, ”lus”, ”ust”, ...]; ”clustering”=[”clu”, ”lus”, ”ust”, ..., ’ing’]; (2) TF-IDF: we measure how frequently one n-gram occurs in a concept, and use TF-IDF to weight the n-gram components; (3) cosine similarity: we compute the cosine similarity of the two concepts; (4) merging: we merge concepts based on their similarity. We tried different thresholds and finally select 0.5 as a proper choice. In addition, there are many abbreviations in the dataset, we manually replace some of them into full forms.

We create an undirected link between two nodes when the corresponding concepts occur together in one paper. We also tested higher thresholds (e.g., establish a link if the concepts co-occur in n papers) but these seemed to make prediction harder.

[3] https://www.aminer.org/citation

B Model Configurations and Training

LP model and CP-gnn use fives graph convolutional neural network layers Kipf & Welling (2017) to generate the node embeddings and five fully connected layers (with dropout 0.5) to obtain the final prediction scores. The hidden dimension is 128 everywhere. And we use the Sigmoid function to convert the final prediction score to 0-1 range. As learning rate we chose 0.002. Note that we did not do extensive hyperparameter tuning. The implementation was done in PyTorch Paszke et al. (2019) and PyTorch Geometric Fey & Lenssen (2019).

CP models use BERT to encode the concepts with fixed prefix ”This paper uses”. We take the CLS token for the downstream task. For CP-pro, CP-avg and CP-gnn, we use two layer fully connected layers with dropout 0.5 to obtain the final prediction scores. For CP-enb, our scoring function can predict the combinatorial scores directly.

During training, we randomly generate target nodes to create negative training samples (i.e., pairs of concepts) for LP. For CP, for each S of more than 4 concepts, we break it down into subsets of shorter combinations $\{S'\}$ for easier training. The negative combination are then produced by randomly replacing one concept for each S'.

We trained both models for maximal 300 epochs with a patience of 30 epochs using three different random seed for each experiment, using F1 for model selection. We use 1e-3 as the learning rate and batch size 256 for LP with 5 GCN layers. For all CP methods, we use ”bert-base-uncased” as our pre-trained language model from Pytorch. We use 1e-5 as our learning rate to train CP-pro, CP-avg and CP-enb. For CP-gnn, the learning rate is 1e-4. NLP dataset requires about 12 hours to train on one GeForce RTX 1080ti GPU and CV dataset requires two and a half day to train. All models will train a binary classifier at the end and we use the threshold 0.5 to decide the prediction is positive or negative.