Economics Assistant for Robustness Checks (EconARC): Identifying Confounders from Causal Knowledge Graphs

Fiona Anting Tan, See-Kiong Ng

Institute of Data Science, National University of Singapore

Abstract

In Economics, authors conduct robustness checks, such as accounting for potential confounders, to avoid drawing misleading conclusions from their causal analyses. To assist in this process, we propose EconARC, a tool to automatically identify confounders from the literature relevant to a Cause and Effect pair. Our methodology involves extracting cause-and-effect arguments using a fine-tuned sequence-to-sequence model, clustering semantically similar arguments into topics, and utilizing the backdoor criterion on the causal graph to detect confounders. Our study is the first to employ text mining techniques to generate confounders in Economics, with implications for advancing Artificial Intelligence towards human-level capabilities like engaging in academic discourse.

Keywords

causal text mining, confounder detection, knowledge graphs, backdoor criterion

1. Introduction

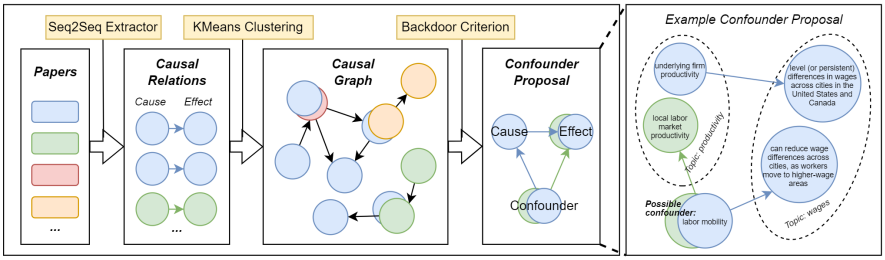

Figure 1: Overview of EconARC

Causal inference relies on addressing confounders, which are variables affecting both the dependent and independent variables. In Economics, consideration of confounders is important, and often a critical part of referee reports. However, staying abreast of confounders in this field requires an extensive knowledge of the literature, which is a non-trivial task given the lengthy and vast number of Economics papers.

ISWC 2023 Posters and Demos: 22nd International Semantic Web Conference, November 6–10, 2023, Athens, Greece
tan.f@u.nus.edu (F. A. Tan); seekiong@nus.edu.sg (S. Ng)

Existing literature on confounder detection from causal knowledge graphs prioritizes uncovering latent relationships using quantitative variables [1]. We offer a novel approach by focusing solely on text, thereby bridging the fields of causal text mining and causal identification. We propose the Economics Assistant for Robustness Checks (EconARC), overview shown in Figure 1, that automates robustness check proposal by identifying confounders related to a Cause and Effect pair. To our knowledge, we are the first work to use causal text mining techniques to generate confounders for Economics. We believe that EconARC will be a useful tool for Economics authors to review their paper prior to submission, and reviewers to obtain an unbiased, initial assessment of a paper. Our work also has implications for advancing Artificial Intelligence (AI) towards human-level understanding and inference tasks like engaging in academic discourse.

2. Our Approach

In this section, we outline our methodology, and provide additional details in the Appendix.[1]

2.1. Dataset

Our study experiments on 177 papers from 23 issues of the Journal of Labor Economics (JOLE).[2] *(1) Annotating Causal Relations:* One of our authors, who is an Econometrics graduate, annotated 5 papers (2,223 sentences) for training and 1 paper (560 sentences) for testing with causal relations. We restricted our annotations to causal relations that appear across 5-sentences in the same section of the paper, and for cause and effect arguments to be consecutive spans. We mainly adapted the annotation guidelines from the Causal News Corpus (CNC) [2, 3] with the key difference being that our annotated causal relations must be helpful to an Economics academic. A consequence of this rule means that we differ from CNC in areas like our arguments need not contain events, and we do not annotate: (1) Purpose relations, (2) justifications for a data or methodology choice, etc. In total, 522 and 76 causal relations were annotated in the training and test set respectively. *(2) Annotating Argument Topics:* For two papers from the training set, the same annotator assigned open-ended topic labels to each argument. 356 arguments were annotated to 119 topics. Topic labels were as general as *"education"*, to more specific labels like *"greater upward mobility"*, *"areas where fathers tend to be richer within the bottom half of households and whose sons did better accordingly"*, etc.

Sequences without annotations will be referred to as our Out-of-Sample (OOS) set. The OOS set comprises of 83,676 sentences from 171 papers.

2.2. Extraction of Causal Relations

We fine-tune SOTA sequence-to-sequence (S2S) pre-trained language models (PTMs), like `t5-base` [4], `bart-base` [5] and `pegasus-large` [6]. Given the `input_text`, the model learned to generate the `target_text`. These texts are described below:

1. `input_text`: An input sequence that is 5-sentences long with a "`summarize: `" prefix.

[1] Our repository is available at https://github.com/tanfiona/EconARC.
[2] https://www.journals.uchicago.edu/toc/jole/current

2. `target_text`: If no causal relations were annotated within the `input_text`, return "No key causal relations". Else, return a line-separated list of causal relations in the format of "Key causal relations:`\n1. Cause: <FIRST_CAUSE_SPAN>\tEffect:<FIRST_EFFECT_SPAN>\n...`"

2.3. Knowledge Graph Creation

To prevent a sparse graph, we grouped arguments with similar meaning into a topic. Similar to [7, 8], we approached this task by (1) generating word embeddings and (2) clustering the embeddings. We concatenated the annotated causal relations from the training set and the inferred causal relations from the OOS set together when performing clustering. For (1), to encode arguments into embeddings, we experimented with PTMs like the supervised pre-trained language model by SimCSE [9] and the encoder portion of our fine-tuned T5 extraction model. For (2), we condensed our embeddings into 400 components[3] using Principal Components Analysis (PCA) and used Mini-Batch K-Means [10] to perform our clustering. We explored various levels of K (5000 to 15000, jumping by gaps of 2500). We removed relations where the cause and effect have the same topic to avoid nodes with self-loops. Our knowledge graph (KG) $G = (V, E)$ is a collection of nodes $V = \{(v_1, v_2, ..., v_n)\}$ and directed edges $E = \{(v_1, v_2), (v_2, v_3), ...\}$. A directed edge (v_x, v_y) represents the presence of causality between the two nodes, where v_x is the cause argument and v_y is the effect argument. The edges are also weighted by support s, indicating the count of relations expressing causality from v_x to v_y in the dataset.

2.4. Confounder Detection

Given an ordered pair of variables (X, Y) in a directed acyclic graph G, a set of variables Z satisfies the **backdoor criterion** relative to (X, Y) if no node in Z is a descendant of X, and Z blocks every path between X and Y that contains an arrow into X [1]. Backdoor paths may make X and Y dependent despite lacking causal influences from X. To estimate the causal relationship of X on Y, we need to condition on a set of nodes Z such that Z (1) blocks all spurious paths between X and Y, (2) leaves directed paths between X and Y unchanged, and (3) creates no new spurious paths. In other words, the causal effect of X on Y is given by the formula: $P(Y = v_y|do(X = v_x)) = \sum_{v_c} P(Y = v_y|X = v_x, Z = v_z)P(Z = v_z)$. This formula describes the distribution of Y given an intervention ($do(X = v_x)$) that sets X to the value v_x, thereby removing X's dependence on Z.

Given a source (Cause) and target (Effect), we automatically identify potential confounders by adapting the backdoor criterion scripts from DoWhy [11], a Python package for causal inference. A depth-first search algorithm to explore paths between the Cause and Effect pair and determines the variables that need to be conditioned on to block all paths between them. Since our whole graph is too large, we had to restrict our search space to improve run times: For each node in the graph G, we designated it as a central node and obtained a subgraph (sG) containing nodes located within a 2-step radius. For each node in sG that is not the central node, we designated it as the target node, while the central node was fixed as the source. The benefit of this setup is that we could search for backdoor variables within a feasible run time. However, our methodology fails to identify backdoor variables that lie outside of each subgraph.

[3]With 400 components, only 5.005^{-05} of variance is dropped.

(A) Extraction (Seq2Seq Model)

PTM	ROUGE1	ROUGE2	ROUGEL	ROUGELsum
T5	**79.90**	**77.65**	**79.25**	**79.65**
Pegasus	66.27	63.07	65.51	65.73
BART	76.86	73.76	75.97	76.48

(B) Clustering (MiniBatch K-Means Model)

PTM	K	ARI	FMI	NMI
SimCSE	7500	**23.03**	**32.95**	**82.47**
SimCSE	10000	18.32	27.51	80.54
T5	7500	15.81	21.53	77.64
T5	10000	12.19	21.50	80.11

Table 1
Performance metrics for (A) cause-effect extraction in a S2S framework and (B) argument clustering using Mini-Batch K-Means. Scores are reported in percentages (%). Top score per column is in bold. Explanations for evaluation metrics are available in the Appendix.

3. Results & Conclusion

Panel A of Table 1 reports scores for extraction. Across all metrics, our best model was the S2S model that fine-tuned T5, scoring 79.90% for ROUGE1 and 76.25% for ROUGEL.[4] Hence, we used this best model on our OOS set to obtain predicted causal relations. Panel B of Table 1 reports scores for clustering. Across all metrics, our best model uses SimCSE embeddings and performs K-Means clustering for 7500 topics, scoring 23.03% for ARI and 82.47% for NMI. Using the SimCSE embeddings consistently supercedes using T5's, suggesting benefits in clustering arguments that were converted to embeddings that convey semantic similarity. For our best model, our KG comprises of 7498 unique nodes, 37557 edges, and an edge support ranging from 1 to 10 and averaging at 1.207.[5] Finally, we apply the backdoor criterion detection algorithm to identify confounders. For our dataset of 176 papers (train + OOS), we identified 152 papers and 676 confounders for authors to consider reviewing. These confounders lie 1 to 4 steps away from either the cause or effect argument of the main relation. We also detected 161 papers and 1408 confounders that the authors themselves describe within their paper, which reveal that confounders and robustness checks are definitely a key concern and covered by most authors.

In conclusion, EconARC successfully applies causal text mining techniques to automatically identify confounders. EconARC will be a useful tool for Economics authors and referees to critically evaluate the validity of a causal identification strategy. This tool will also help mitigate reviewers' unconscious bias by standardizing the review process. In the future, we hope to expand the coverage of our work to more journals and to more branches of Economics, and to evaluate our system with Economic academics. We also hope to design tools to identify other threats to validity to provide a more comprehensive review.

References

[1] J. Pearl, M. Glymour, N. P. Jewell, Chapter 3: The effects of interventions, in: Causal inference in statistics: A primer, John Wiley & Sons, 2016.

[2] F. A. Tan, A. Hürriyetoğlu, T. Caselli, N. Oostdijk, T. Nomoto, H. Hettiarachchi, I. Ameer, O. Uca, F. F. Liza, T. Hu, The causal news corpus: Annotating causal relations in event

[4] We used ROUGE evaluation metrics since the task is a S2S open-ended generation task.

[5] Due to limited space, we provide experimental details, explanation of our evaluation metrics, ablation studies and provide qualitative examples of confounders in the Appendix.

sentences from news, in: Proceedings of the Thirteenth Language Resources and Evaluation Conference, European Language Resources Association, Marseille, France, 2022, pp. 2298–2310. URL: https://aclanthology.org/2022.lrec-1.246.

[3] F. A. Tan, H. Hettiarachchi, A. Hürriyetoğlu, N. Oostdijk, T. Caselli, T. Nomoto, O. Uca, F. F. Liza, S.-K. Ng, RECESS: Resource for extracting cause, effect, and signal spans, in: Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, Bali, Indonesia, 2023.

[4] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, Journal of Machine Learning Research 21 (2020) 1–67. URL: http://jmlr.org/papers/v21/20-074.html.

[5] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer, BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 7871–7880. URL: https://aclanthology.org/2020.acl-main.703. doi:10.18653/v1/2020.acl-main.703.

[6] J. Zhang, Y. Zhao, M. Saleh, P. J. Liu, PEGASUS: pre-training with extracted gap-sentences for abstractive summarization, in: Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event, volume 119 of *Proceedings of Machine Learning Research*, PMLR, 2020, pp. 11328–11339. URL: http://proceedings.mlr.press/v119/zhang20ae.html.

[7] S. Sia, A. Dalmia, S. J. Mielke, Tired of topic models? clusters of pretrained word embeddings make for fast and good topics too!, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 1728–1736. URL: https://aclanthology.org/2020.emnlp-main.135. doi:10.18653/v1/2020.emnlp-main.135.

[8] Z. Zhang, M. Fang, L. Chen, M. R. Namazi Rad, Is neural topic modelling better than clustering? an empirical study on clustering with contextual embeddings for topics, in: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Seattle, United States, 2022, pp. 3886–3893. URL: https://aclanthology.org/2022.naacl-main.285. doi:10.18653/v1/2022.naacl-main.285.

[9] T. Gao, X. Yao, D. Chen, SimCSE: Simple contrastive learning of sentence embeddings, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2021, pp. 6894–6910. URL: https://aclanthology.org/2021.emnlp-main.552. doi:10.18653/v1/2021.emnlp-main.552.

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[11] A. Sharma, E. Kiciman, Dowhy: An end-to-end library for causal inference, CoRR abs/2011.04216 (2020). URL: https://arxiv.org/abs/2011.04216. arXiv:2011.04216.