Causal DAG extraction from a library of books or videos/movies

Robert R. Tucci

tucci@ar-tiste.com

October 19, 2022

Abstract

We argue that human and animal brains contain an explicit engine for doing Causal Inference, and that such an engine uses as input an atlas (i.e., collection) of causal DAGs. We propose a simple algorithm for constructing such an atlas from a library of books or videos/movies. We illustrate our method by applying it to a randomly generated dataset of Tic-Tac-Toe games.

"If humans were so good at causal inference, religion would not exist."

- Yann LeCun, *Ref.[1]*

"How much of human knowledge is captured in all the text ever written? To which my answer is: not much. "

- Yann LeCun, *Ref.[2]*

1   Introduction

Automated Causal DAG Extraction From Text (DEFT) seems like a daunting task today, at a time when little work has been done to achieve it. Some previous work to achieve DEFT can be found in Ref.[5]. Once DEFT is achieved, the extracted DAGs can be used as inputs for a Causal Inference (CI) engine within AI software. The job of such a CI engine will be to distinguish between correlation and causation, and to perform all 3 rungs (prediction, do operators, counterfactuals) of Judea Pearl's ladder of CI. (See Ref. [3]).

Contrary to what naysayers like Yann LeCun predict, we believe an explicit CI engine will some day be an essential part of all AI software.

We also believe that, contrary to what Yann LeCun claims in the above quote, there is ample evidence that a very efficient CI engine already exists in all human and animal brains. Maybe not all human brains (e.g., Yann LeCun's) excel at CI, but some clearly have (e.g., Einstein's and Feynman's).

Some people, like Yann LeCun, believe that human/animal brains, rather than possessing an explicit CI engine, perform CI through spontaneous generation, or random chance, or Divine Intervention. We vehemently disagree. An AI without an explicit CI engine would be DIM (Divine Intervention Mechanism) witted, and would not act like the human or animal mind. If at the dawn of life, human and animal brains had lacked an explicit CI engine, Nature, through evolution, would have invented it early on. The survival advantage of such a CI engine is enormous.

Finally, contrary to what LeCun claims in the above quote, the knowledge captured in all the text ever written, is enormous. One of the goals of this paper is to improve AI by mining the vast trove of human knowledge (especially, causal knowledge) contained in text (and other sources like videos). Once that vast trove of human knowledge can be mined in an automated fashion, this will open the floodgates for implementing CI in AI. As Judea Pearl has said, CI will elevate AI from Deep Learning to Deep Understanding.

The DEFT algorithm proposed in this paper extracts causal DAGs. This begs the question, what is causality? This is how I define causality in my book "Bayesuvius" (Ref.[4])

Causality is a time-induced ordering between two events, the transmission of information (and its accompanying energy) from the earlier of the two events to the later one, and the physical response of the later event to the reception of that information.

I expand more on this definition in Bayesuvius, in its Appendix E entitled: "Bayesian Networks, Causality and the Passage of Time".

2 Simple proposal for a DEFT algorithm

In this section, we will present our proposal for a simple DEFT algorithm. We will use the acronym DEFT to mean extraction of DAGs from various primary sources. By **primary sources**, we will mean, not just text (i.e., libraries of books), but also libraries of videos and other sources or a mixture of these.

The first step in our algorithm is to covert the primary sources into a library of one or more comic books (CBs). By a **comic book (CB)**, we mean a book that consists of a finite number of chronologically ordered **frames**. Different CBs in the library may contain a different number of frames. Each frame in a CB must be annotated with an **event descriptor** for each of the most important **events** occurring in the frame. An example of an event descriptor might be "Boy kicks soccer ball". It's just a very simple sentence with at least a subject and a verb, and, occasionally, more info than a subject-verb.

Reduction of primary sources to CBs might be done completely by machine learning software (fully-automated) or completely by humans or both. There already exist various image-to-text and text-to-image conversion software apps. If the primary sources are a library of movies, perhaps the image-to-text conversion apps can be modified to select a finite number of chronological ordered frames and to annotate them with event descriptors. If the primary sources are a library of books, perhaps the text-to-image apps can be modified to produce the CBs. Hence, it's possible that someday a fully automated DEFT algorithm will be achieved.

The reason why we do primary sources to CBs conversion as the first step of our DEFT algorithm is that a primary source like a text often has the chronological order of events implicit. To mine the causal info of a primary source, we first need to make its chronological order of events very clear and explicit. There is much more to causality than chronological ordering, but chronological ordering is an intrinsic component of it.

The second step in our DEFT algorithm is to convert the library of CBs produced in the first step into a collection of DAGs (**DAG atlas**). We shall describe this step using a simple fictional example.

For our fictional example, we shall end up considering a **number of events** $N_e = 11$. The event descriptors are as follows. Note that sometimes, to save space, we will abbreviate the event descriptor by a 4 letter symbol.

1. *SuRi:* Sun rises

2. *RoCr:* Rooster crows

3. *BuSt:* Bus starts daily rounds

4. *BoWa:* Boy wakes up

5. *BoEa:* Boy eats breakfast

6. *BoRu:* Boy runs to meet bus

7. *BuAr:* Bus arrives at boy's house

8. *DoBa:* Dog barks

9. *BoBo:* Boy boards bus

10. *RaSt:* Rain starts

11. *RaEn:* Rain ends

Supoose two CBs, say c_1 and c_2, have event sets $E(c_1)$ and $E(c_2)$, respectively. We say c_1 and c_2 are **time compatible** if either of the following two possibilities hold: (1) $E(c_1) \subset E(c_2)$ and the events in $E(c_1)$ are in the same chronological order as the corresponding events in $E(c_2)$. (2) The same as (1) but with c_1 and c_2 interchanged. This obvious defines an equivalence relation $c_1 \sim c_2$.

We define a **time compatible library (tclib)** of CBs to be a library of CBs wherein all the CBs are mutually time compatible.

A **CB c is time compatible with a tclib** C of CBs if c is time compatible with every CB in C.

If our entire library of CBs consists of just one CB called c, then create a new tclib with c as its only member. If we have a collection tclibs $\mathcal{C}$ already, and we want to add a new CB called c, to the collection, we go through each tclib in $\mathcal{C}$ until we find a tclib with which c is time compatible. If we can't find a tclib that fits, we create a new tclib with c as its only member.

Next we shall explain how to process each tclib individually.

Let $\gamma = 1, 2, 3, \ldots, N_{cb}$ label the CBs in our current tclib of CBs. For the γ'th CB, we construct a DAG G_γ. Fig.1 shows, for our fictional example, the DAGs for the first 3 CBs of our current tclib of CBs. The DAGs in Fig.1 were constructed very easily by placing the event descriptors in chronological order with time pointing down the page, and adding arrows entering each event e from all events e' in the frame immediately before e's frame. We will refer to this as $N_{mem} = 1$ **memory time**. We could have just as well added arrows entering each event e from all events e' in the two frames immediately before e's frame. This would be the case $N_{mem} = 2$. In general, one could use any N_{mem}, including N_{mem} as big as the total number of
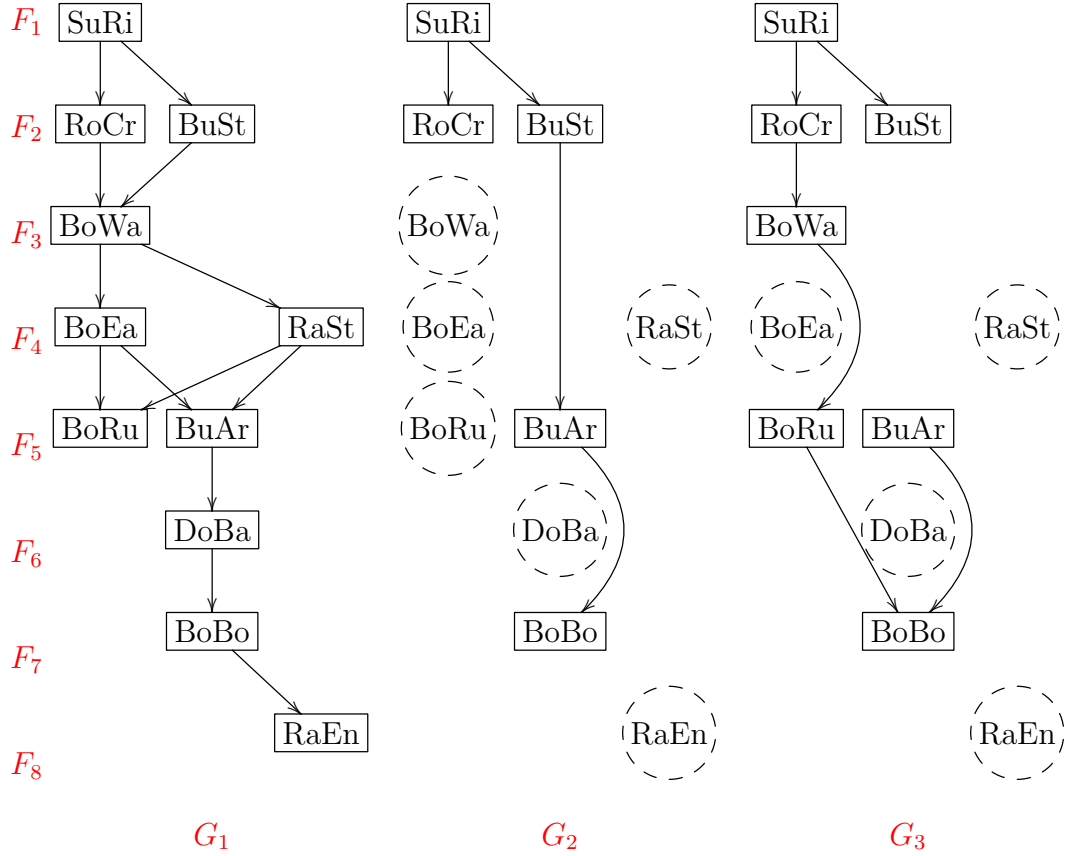
Figure 1: Let $\gamma = 1, 2, 3, \ldots, N_{cb}$ label the CBs in our current tclib of CBs. For the γ'th CB, we construct a DAG G_γ. Here we show the DAGs for the first 3 CBs of our current tclib of CBs. Each row F_i for $i = 1, 2, \ldots, N_f$ are the frames of the CBs. Nodes enclosed by a rectangle represent events that happened in a CB. Nodes enclosed by a dashed circle are placeholders for events that didn't happen in a CB.

frames, but the higher N_{mem} is, the more dense the DAG will be. I think N_{mem} equal to 1 or 2 will provide a good enough causal fit most of the time. The choice of N_{mem} also depends on the average number of event descriptors per frame of the CB.

At this point, we are ready to calculate G_{hfa}, the **high frequency arrows (hfa)** DAG. This DAG is created by counting the number of times each arrow appears in the just created DAGs G_γ for all γ, and keeping only arrows that are repeated at least as much as the **arrow repetition threshold** N_{art}. Fig.2 shows a possible G_{hfa} for our fictional example, assuming $N_{art} = 50$. The intuition behind G_{hfa} is that arrows that have low repetition numbers do so because Nature is telling us that they are not essential for the causal mechanism being expressed by the whole DAG.

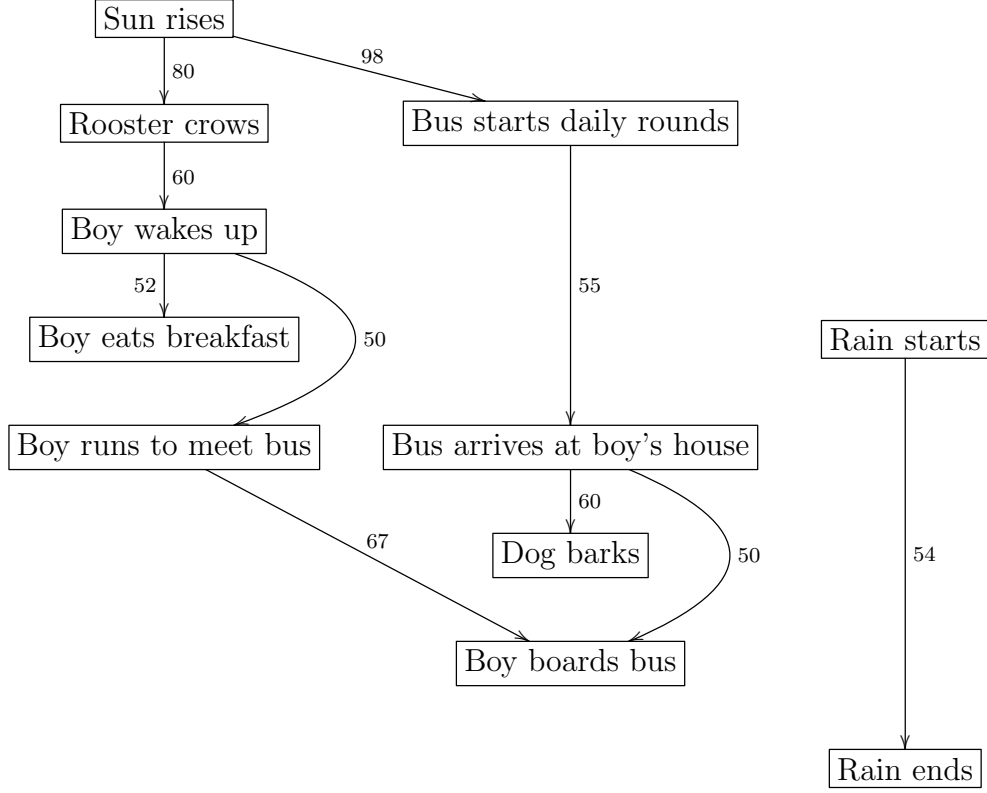Note that one can upgrade G_{hfa} from a causal DAG to a causal Bayesian Net-

Figure 2: High frequency arrows DAG, G_{hfa}, with $N_{art} = 50$. Arrow repetition number given besides each arrow. Fig.1 shows DAGs G_γ for $\gamma = 1, 2, 3$.

work (bnet) by calculating an empirical estimate for the TPM (Transition Probability Matrix) of each event node $\underline{e} \in \{0, 1\}$ of G_{hfa}. This estimate can be calculated using the following formula.

$$P(e|pa(\underline{e}) = \pi) = \frac{\sum_{\gamma=1}^{N_{cb}} \mathbb{1}(\underline{e}_\gamma = e, [pa(\underline{e})]_\gamma = \pi)}{\sum_{\gamma=1}^{N_{cb}} \mathbb{1}([pa(\underline{e})]_\gamma = \pi)} \tag{1}$$

Here $e \in \{0, 1\}$ and $\pi \in \{0, 1\}^n$, where n is the number of parents of node $\underline{e}$ of G_{hfa}. Also, node $\underline{e}$ in G_{hfa} and node $\underline{e}_\gamma$ in G_γ refer to the same event.[1]

Note that G_{hfa} might consist of more than one connected DAG. Furthermore, note that we have shown how to generate a G_{hfa} for each tclib in a collection $\mathcal{C}$ of tclibs. Hence, this algorithm generates a collection of G_{hfa} DAGs. This collection of DAGs is what we call the **DAG atlas**.

Note that our DEFT algorithm is very simple and only requires keeping good chronologically ordered records and counting. We believe the human brain excels at

---

[1]We are using the convention used in Bayesuvius (Ref.[4]) of indicating random variables by underlining them, and using $pa(\underline{x})$ to denote the parent nodes of node $\underline{x}$.

those tasks. Our DEFT algorithm avoids performing precise numerical calculations like calculating π to 15 decimal places because we believe most human brains can't perform such tasks well.

3   How to use the DAG Atlas

Once a DAG atlas is calculated, and its DAGs are upgraded to bnets, it can be used to perform predictions (1st rung), do operators (2nd rung), and to calculate CATE and ATE for counterfactuals (3rd rung). The results of these calculations can be used by the AI to make decisions.

In the particular version of the 3 rungs that is advocated in Bayesuvius [2], the 2nd rung removes arrows from a bnet using a do-operator, and the 3rd rung adds new nodes and arrows to a bnet using an imagine operator. My point is that the DAG atlas can be modified by applying the rules of CI.

Note that hidden nodes have not been mentioned so far. Recall that hidden nodes are nodes in a DAG that one would like to include because we think they are important, but which haven't been measured, either because it is impossible to measure them, or we simply haven't gotten around to measuring them. Hidden nodes could certainly be added to the DAGs of a DAG altas, either in an automated fashion, or by hand.

Note that if a CB repeats a task over and over again, the DAG G_{hfa} might turn out to be a dynamical Bayesian network.[3]

A   Appendix: Tic-Tac-Toe example

References

[1] Yann LeCun. Twitter, all world text quote. `https://twitter.com/ylecun/status/1562137291845521408`.

[2] Yann LeCun. Twitter, causal inference and religion quote. `https://twitter.com/ylecun/status/1577128801620070400`.

[3] Judea Pearl and Dana Mackenzie. *The book of why: the new science of cause and effect.* Basic books, 2018.

[4] Robert R. Tucci. Bayesuvius (book). `https://github.com/rrtucci/Bayesuvius/raw/master/main.pdf`.

---

[2]See the chapter entitled "Counterfactual Reasoning" in Bayesuvius, Ref. [4]
[3]Dynamical Bayesian networks are discussed in Bayesuvius, Ref.[4]

[5] Justin Wood, Nicholas Matiasz, Alcino Silva, William Hsu, Alexej Abyzov, and Wei Wang. Opberg: Discovering causal sentences using optimal alignments. In *International Conference on Big Data Analytics and Knowledge Discovery*, pages 17–30. Springer, 2022.