Explainable Bayesian Network Query Results via Natural Language Generation Systems

Jeroen Keppens*
jeroen.keppens@kcl.ac.uk
King's College London
London, United Kingdom

ABSTRACT

Bayesian networks (BNs) are an important modelling technique used to support certain types of decision making in law and forensics. Their value lies in their ability to infer the rational implications of probabilistic knowledge and beliefs, a task that human decision makers struggle with. However, their use is controversial. One of the main obstacles to the more widespread use of BNs is the difficulty to acquire good explanations of the results obtained with BNs. While useful techniques exist to visualise, verbalise or abstract BNs and the inner workings of belief propagation algorithms, these techniques provide generic, one-size-fits-all explanations, that have, thus far, failed to stem the criticism of lack of explainable BN results. Building on the qualified support graph method introduced in earlier work, this paper outlines how a natural language generation system can be constructed to explain Bayesian inference. This constitutes a novel approach to BN explanation that has the potential to produce more focussed and compelling explanations of Bayesian inference as the narratives such a system produces can be tailored to address specific communicative goals and, by extension, the needs of the user.

CCS CONCEPTS

• **Applied computing** → **Law**; • **Computing methodologies** → **Natural language generation**; **Probabilistic reasoning**; • **Mathematics of computing** → **Bayesian networks**;

ACM Reference format:
Jeroen Keppens. 2019. Explainable Bayesian Network Query Results via Natural Language Generation Systems. In *Proceedings of Seventeenth International Conference on Artificial Intelligence and Law, Montreal, QC, Canada, June 17–21, 2019 (ICAIL '19),* 10 pages.
https://doi.org/10.1145/3322640.3326716

1 INTRODUCTION

Bayesian networks (BNs) are an important decision support tool for evidential reasoning in law. They allow for complex scenarios to be modelled – incorporating expert knowledge and opinion – and analysed by computing the rational implications of the formalised knowledge and opinions. BNs are used to calculate the value of evidence in evaluating hypothesised scenarios [1], the probability of individual hypotheses [9] and directing forensic inquiry [7]. Applications are very diverse and include assessing the value multiple pieces of trace evidence [2, 10], profiling DNA evidence taken from multiple sources [17], forensic investigation of fire incidents [5], traffic accident reconstruction [8] and evaluating the possible causes of back injury at work [12].

But the use of BNs, and probabilistic approaches in general, in law is controversial. One reason for this is that Bayesian approaches have been misapplied or misused in numerous high-profile cases, sometimes leading to gross miscarriages of justice. Another is that probabilistic reasoning is notoriously challenging for human beings. Correct results can seem counter-intuitive, even wrong, to the human decision makers who are required to work with these models but struggle to understand whether their surprise at seeing certain results stems from errors in the model or their own misconceptions.

A wide range of tools have been designed to explain the inference that takes place in a Bayesian network [14]. Typically, these tools visualise, verbalise or abstract the BN model or the inner workings of the belief propagation algorithm. While many of these tools provide very helpful clarifications, and some have gained widespread adoption[1], they tend to require a good understanding of the model being explained and, therefore, they do not work well as stand-alone accounts of Bayesian inference.

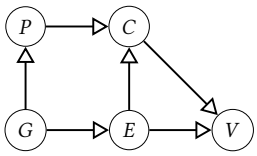
This paper aims to contribute towards an approach to produce natural language explanations of belief propagation in a BN that can be understood without the BN model. Specifically, this paper sets out to show how such a system can be constructed, what components it ought to consist of and the inputs and outputs each component requires. The approach proposed in this paper extends the support graph formalism introduced in earlier work [13, 22] to produce a comprehensive content model from which narrative explanations can be generated by means of natural language generation (NLG) techniques. The paper then goes on to explain how NLG techniques can be applied in this context. Throughout, the ideas introduced are illustrated by means of the Jury observation fallacy BN developed by Fenton and Neil [9].

The natural language explanation system proposed herein is envisioned to produce explanations that help empower human decision makers to communicate about a BN and the results obtained with a BN by helping them understand how a result was arrived at. Such improved understanding may either convince them that a result is correct, or help them ask critical questions of aspects of

[1] Most Bayesian network tools, such as Hugin [3] and AgenaRisk [9], include a basic explanation facility, that consist of an interactive visualisation of the network and the posterior probability distributions of variables within it.

ICAIL '19, June 17–21, 2019, Montreal, QC, Canada

ACM ISBN 978-1-4503-6754-7/19/06…$15.00
https://doi.org/10.1145/3322640.3326716



g	the defendant is guilty
$\overline{g}$	the defendant is not guilty
p	the defendant committed prior offences
$\overline{p}$	the defendant did not commit prior offences
e	there is hard evidence supporting the defendant's guilt
$\overline{e}$	there is no hard evidence supporting the defendant's guilt
c	the defendant is charged
$\overline{c}$	the defendant is not charged
v	the defendant is found guilty
$\overline{v}$	the defendant is found not guilty
v_0	there is no trial

$Pr(g)$	0.0001
$Pr(\overline{g})$	0.9999

G	g	$\overline{g}$
$Pr(p\|G)$	0.1	0.0001
$Pr(\overline{p}\|G)$	0.9	0.9999

G	g	$\overline{g}$
$Pr(e\|G)$	0.95	0.000001
$Pr(\overline{e}\|G)$	0.05	0.999999

E	e		$\overline{e}$	
P	p	$\overline{p}$	p	$\overline{p}$
$Pr(c\|P,E)$	0.9999999	0.99	0.02	0.00001
$Pr(\overline{c}\|P,E)$	0.0000001	0.01	0.98	0.99999

C	c		$\overline{c}$	
E	e	$\overline{e}$	e	$\overline{e}$
$Pr(v\|C,E)$	0.99	0.01	0	0
$Pr(\overline{v}\|C,E)$	0.01	0.99	0	0
$Pr(v_0\|C,E)$	0	0	1	1

Figure 1: Jury observation fallacy BN (from [9]). The node probability tables were proposed as an example by the original authors.

the model that played a significant role in achieving that result. It is anticipated that, in the application of BNs in Law, where non-experts are often expected to engage in a meaningful manner with the mathematical models produced by experts, a means to facilitate communication about such models would be welcomed.

2 BACKGROUND

A Bayesian network (BN) is a type of probabilistic model used to describe scenarios where multiple stochastic variables are related with one another. Figure 1 presents an example of a BN developed by Fenton and Neil [9] to analyse the Jury observation fallacy problem. The scenario being analysed here is one where a Jury has decided that the defendant in a case is not guilty of the crime with which he was charged. Afterwards, a juror reads a newspaper article noting that the defendant had been convicted of similar prior offence, information that was withheld from the Jury during Court proceedings. As a result of this information, she feels it is now more likely that the defendant was guilty after all. This scenario of juror's remorse is not uncommon, though not necessarily justified.

Figure 1 models this problem by means of five variables and a directed acyclic graph (DAG) specifying how these variables are related. The defendant's guilt (G) affects the probability that hard evidence demonstrating the defendant's culpability is collected (E) and, to a lesser extent the probability that the defendant has committed prior offences (P). Both these variables affect the probability that the defendant is charged (C). If and only if a defendant is charged can a guilty or not guilty verdict be returned (V), and the probability of this is affected by the availability of evidence.

Formally, a BN is specified by means of a directed acyclic graph (DAG) $\langle \mathbf{V}, \mathbf{E} \rangle$, where $\mathbf{V}$ is a set of vertices and $\mathbf{E}$ is a set of directed edges between pairs of variables in $\mathbf{V}$, and a set of node probability tables, one for each vertex [19]. Each vertex $V \in \mathbf{V}$ corresponds to a stochastic variable with a domain of mutually exclusive values $\mathrm{Dom}(V)$. As such, the term vertex and variable of a BN can be used interchangeably. The situation in any possible world is described by assigning each variable V in the BN exactly one of the values $v_i \in \mathrm{Dom}(V)$ of its domain (hereafter denoted $V : v_i$). The example BN of Figure 1 consists of five variables $\mathbf{V} = \{G, P, E, C, V\}$, where G, P, E, C are all Boolean variables representing that a proposition is either true or false and $\mathrm{Dom}(V) = \{v, \overline{v}, v_0\}$. In the example $\mathbf{E} = \{G \dashrightarrow P, G \dashrightarrow E, P \dashrightarrow C, E \dashrightarrow C, E \dashrightarrow V, C \dashrightarrow V\}$.

The edges of the BN define conditional independence relations between variables. Let the parents of a variable V be the set of variables directly connected by an edge to V ($\mathrm{Par}(V) = \{V_p \in \mathbf{V} | V_p \dashrightarrow V \in \mathbf{E}\}$) and the descendents of a variable V be the set of variables to which $\mathbf{E}$ defines a directed path from V ($\mathrm{Desc}(V) = \{V_d \in \mathbf{V} | \exists P \dashrightarrow V \in \mathbf{E}, P \in \mathrm{Par}(V) \vee P \in \mathrm{Desc}(V)\}$). The set of edges of a BN imply that any variable V is probabilistically independent of its non-descendents if all of its parents have a known value [11].

Thanks to this property, the probability distribution of any variable in a BN only needs to be conditioned on its parents. Therefore, each variable V in a BN comes with a node probability table that defines a probability distribution for V for each combination of value assignments to V's parents. For example, the BN of Figure 1 includes four probability distributions for C, one for each combination of values of its parents variables $\mathrm{Par}(C) = \{E, P\}$: i.e. $Pr(C|p, e)$, $Pr\left(C|\overline{p}, e\right)$, $Pr(C|p, \overline{e})$ and $Pr\left(C|\overline{p}, \overline{e}\right)$. These probability distributions may be derived from data, from the author's beliefs, or from a combination of both. In the example, the distribution specifies the beliefs (of the authors of the BN) that (i) it is very likely that the defendant is charged if there is hard evidence supporting the defendant's guilt ($Pr(c|P, e) \geq 0.99$ for any value of P), (ii) it is unlikely that the defendant is charged if there is no hard evidence supporting the defendant's guilt ($Pr(c|P, \overline{e}) \leq 0.02$ for any value of P) and (iii) the defendant is only slightly more likely

to be charged with the offence if (s)he committed prior offences ($Pr(c|p, E)$ is slightly higher than $Pr(c|\overline{p}, E)$ for any value of E).

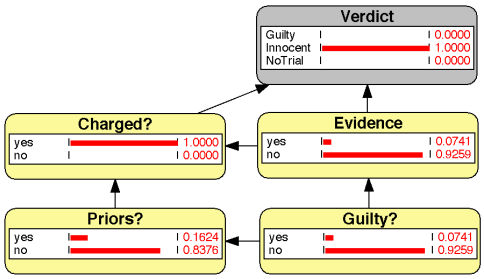


Figure 2: Jury observation fallacy BN, given $V : \overline{v}$

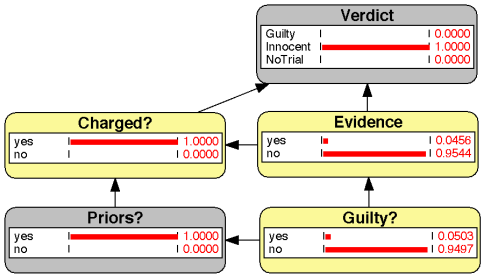


Figure 3: Jury observation fallacy BN, given $V : \overline{v}$ and $P : p$

By modelling our beliefs by means of a BN, such as that shown in Figure 1, we can compute its rational implications and verify whether these correspond to the common intuitive belief that observation of prior offences of a defendant who was already deemed innocent of a crime, increases the probability of the defendant's guilt. Generally speaking, the purpose of the exercise is to assess the effect of an observation o of variable $O \in \mathbf{V}$ on a variable of interest $V \in \mathbf{V}$, given a set of observations (or context) $\mathbf{C}$. Figure 2 show the probability distributions of G, P, E and C at the time the Jury delivers its not-guilty verdict, but before jurors are aware that the defendant committed prior offences. Observe the prior probability of guilt $Pr(g|\overline{v}) = 0.07$. Figure 3 shows the probability distributions of G, E and C after the juror also becomes aware of the defendants prior offences. Observe the posterior probability of guilt is now $Pr(g|\overline{v}, p) = 0.05$. In other words, the probability of guilt has decreased (not increased) as a result of observing that the defendant committed prior offences. Thus, the fallacious reasoning here is that, if the Jury observation fallacy BN is a correct model of the juror's beliefs, knowledge of prior offences should strengthen her belief in the defendant's innocence, not weaken it.

This paper adopts the following notations and assumptions. A BN is fully specified by a tuple $\langle \mathbf{V}, \mathbf{E}, Pr \rangle$. A query is defined by a tuple $\langle V, O : o, \mathbf{C} \rangle$, where V corresponds a variable of interest, $O : o$ is the observation whose effect on V must be assessed and $\mathbf{C}$ is a set of prior observations (variable-value assignments) of variables other than V and O. This paper assumes that all unobserved variables correspond to propositions and have a Boolean domain. Thus, any variable X comes with a domain $\{x, \overline{x}\}$, stating that proposition X is true or false respectively. This simplifies the description of variables and influences between variables considerably, allowing the paper to focus on other challenges in generating explanations of Bayesian inference. Note that in typical applications of BNs in law, namely evaluating the value of evidence, the vast majority of variables used are Booleans and BNs with only Boolean variables are very common.

3 ARCHITECTURE OF A NATURAL LANGUAGE GENERATION SYSTEM

Conventional approaches to explanation of BNs or inference within a BN take the direct approach of translating a model into text [14]. However, natural language generation (NLG) is a complex task that aims to present certain content as natural language text that achieves certain communicative goals. It involves both higher-level reasoning tasks, such as identifying the content to be communicated and structuring it into a narrative, and lower-level reasoning tasks, such as forming sentence structure, applying grammatical rules (e.g. conjugating verbs, pluralising nouns, etc.) and adding punctuation. As BNs are complex models and inference within a BN is a complex process, the generation of effective explanations of BNs does not lend itself to a single stage process.

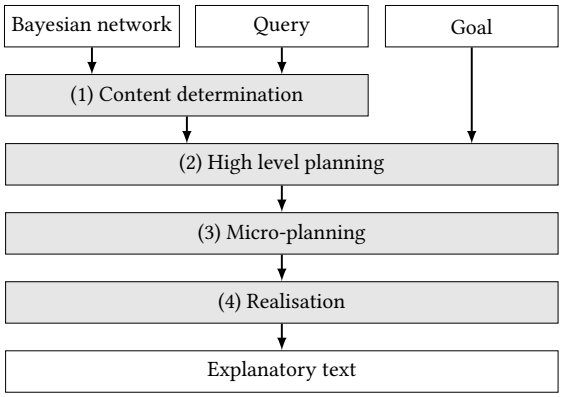


Figure 4: Pipeline architecture of the explanation system

This paper adopts the pipeline architecture approach shown in Figure 4, which is similar to the architecture of many NLG systems. It consists of 4 steps. The content determination subsystem computes an explanatory model of the result of the query of the BN. This model is called the qualified support graph. It is fed to a high level planning system, along with a goal to identify which parts of the qualified support graph must be explained and in what order. The micro-planning subsystem produces a set of sentence structures that implement the high-level plan. Finally, the sentence structures are implemented by the realisation subsystem. The discussion that follows focusses on subsystems (1)-(3) as the realisation module the system requires is not fundamentally different from existing NLG systems.

4 CONTENT DETERMINATION

The first step in a natural language generation system is content determination: identification of the information that is to be communicated through generated text [4, 20]. The approach starts from a given BN and query $\langle V_{\text{query}}, O : o, \mathbf{C}\rangle$. It consists of two steps. First, a graph is generated that identifies all the ways in which observation $O : o$ affects variable V_{query} under context $\mathbf{C}$. The qualified support graph approach proposed by Keppens [13] is used for that purpose here. Next, the support graph is annotated with an assessment of the effect of inferences in the support graph. At this stage, the content determination system assesses the direction and magnitude of inference stems and extends the support graph with the content that is required to produce explanations of the inference steps.

4.1 Support graph generation

The support graph is a concept introduced by Timmer et. al. designed to extract explanatory arguments from a Bayesian network [22]. It represents the ways in which information or support can flow through a Bayesian network: e.g. from evidence to hypothesis nodes. In essence, it provides an alternative view of the notion of d-separation in a BN, which is a more general conception of all conditional independence relationships that exist in a BN. D-separation is perhaps most easily understood in the context of the most basic ways in which three variables X, S and Y can be related by a DAG: a serial connection of the form $X \rightarrow S \rightarrow Y$, a diverging connection of the form $X \leftarrow S \rightarrow Y$ and a converging connection of the form $X \rightarrow S \leftarrow Y$, such that there is no edge between X and Y in any of these three substructures. In a serial connection $X \rightarrow S \rightarrow Y$, X and Y are probabilistically independent from one another provided the value of S is known. Thus, in such a situation, the algorithm that generates the support graph propagates support between X and Y, via S, provided S is not observed. In a diverging connection $X \leftarrow S \rightarrow Y$, X and Y are also probabilistically independent from one another provided the value of S is known. Thus, in such a situation, the algorithm propagates support between X and Y, via S, provided S is not observed. In a converging connection $X \rightarrow S \leftarrow Y$, X and Y are probabilistically independent from one another if the values of S and all of its descendants are *not* known. In other words, X and Y are probabilistically *dependent* from one another, provided the value of S or one of its descendants is known. Here, the algorithm propagates support between X and Y if S or one of its descendents has been observed, but never via S.

Keppens introduced a variant of the support graph method to describe the ways in which a new observation affects a variable of interest [13]. Algorithm 4.1 is a formal presentation in pseudocode of this approach. Given a BN $b = \langle \mathbf{V}, \mathbf{E}, Pr\rangle$ and a query $q = \langle V_{\text{query}}, O : o, \mathbf{C}\rangle$, it computes a support graph $\langle N, L\rangle$ by, starting from $O : o$, plotting all the inferences through which this observation flows through the network to affect the probability distribution of the variable of interest V, in the context $\mathbf{C}$.

This algorithm differs from that of Timmer in three respects. Firstly, the algorithm employs a context $\mathbf{C}$, corresponding to prior observations. An inference is only added to the support graph if the flow of information through it is not blocked by the context, through d-separation (i.e. if the variables related by inference are not independent from one another, given $\mathbf{C}$). Secondly, the algorithm incorporates nodes corresponding to inferences, as these are important in explaining the result of a BN query. There are two types of inference node: (i) influences, which are inferences of the type $V_{\text{parent}} \dashrightarrow V_{\text{child}}$ and (ii) product synergies, which are inferences of the type $V_1 \dashrightarrow V_{\text{child}} \dashleftarrow V_2$ (where V_{child} or at least one

Algorithm 4.1: generateSupportGraph(b, q)

input: A Bayesian network $b = \langle \mathbf{V}, \mathbf{E}, Pr\rangle$
input: A query $q = \langle V_{\text{query}}, O : o, \mathbf{C}\rangle$
output: A support graph $s = \langle N, L\rangle$
procedure extendSupportGraph(V, F)
 for each $V_{\text{parent}} \dashrightarrow V \in \mathbf{E}, (V_{\text{parent}} \notin F) \wedge (\nexists v, V_{\text{parent}} : v \in \mathbf{C})$
  do addParent(V_{parent})
 for each $V \dashrightarrow V_{\text{child}} \in \mathbf{E}, (V_{\text{child}} \notin F) \wedge (\nexists v, V_{\text{child}} : v \in \mathbf{C})$
  do addChild(V_{child})
 for each $V \dashrightarrow V_{\text{child}}, V_{\text{coparent}} \dashrightarrow V_{\text{child}} \in \mathbf{E}, (V_{\text{coparent}} \neq V)$
 $\wedge (V_{\text{child}} \notin F) \wedge (V_{\text{coparent}} \notin F) \wedge$
 $\exists V_{\text{obs}} : v \in \mathbf{C} \cup \{O : o\}, V_{\text{obs}} \in \{V_{\text{child}}\} \cup \text{Desc}(G, V_{\text{child}})$
  do addCoparent$(V, V_{\text{child}}, V_{\text{coparent}}, F)$

procedure addParent$(V, V_{\text{parent}}, F)$
 $i \leftarrow$ getOrCreateInfluence$(V_{\text{parent}} \dashrightarrow V)$
 addToSupportGraph(V, i, V_{child})
 extendSupportGraph$(V_{\text{parent}}, F \cup \{V_{\text{parent}}\})$

procedure addChild(V, V_{child}, F)
 $i \leftarrow$ getOrCreateInfluence$(V \dashrightarrow V_{\text{child}})$
 addToSupportGraph(V, i, V_{child})
 $F_{\text{parents}} \leftarrow \{V_{\text{parent}} | V_{\text{parent}} \dashrightarrow V_{\text{child}} \in \mathbf{E}\}$
 extendSupportGraph$(V_{\text{child}}, F \cup \{V_{\text{child}}\} \cup F_{\text{parents}})$

procedure addCoparent$(V, V_{\text{child}}, V_{\text{coparent}}, F)$
 $i \leftarrow$ getOrCreateSynergy$(V_{\text{coparent}} \dashrightarrow V_{\text{child}} \dashleftarrow V)$
 addToSupportGraph(V, i, V_{child})
 extendSupportGraph$(V_{\text{child}}, F \cup \{V_{\text{child}}, V_{\text{coparent}}\})$

procedure addToSupportGraph(V, i, V_{new})
 $n \leftarrow$ getVariable$((V))$
 $n_{\text{new}} \leftarrow$ getOrCreateVariable(V_{new})
 $N \leftarrow N \cup \{n_{\text{new}}, i\}$
 $L \leftarrow L \cup \{n \rightarrow i, i \rightarrow n_{\text{new}}\}$

procedure prune()
 for each $V_{\text{leaf}} \in N, V_{\text{leaf}} \neq V_{\text{query}}, \nexists V_{\text{leaf}} \rightarrow I \in L$
  do $\begin{cases} N_{\text{diff}} \leftarrow \{I | I \in N, I \rightarrow V_{\text{leaf}} \in L\} \\ L_{\text{diff}} \leftarrow \{I \rightarrow V_{\text{leaf}} \in L\} \cup \{V \rightarrow I \in L | I \in N_{\text{diff}}\} \\ N \leftarrow N \setminus N_{\text{diff}} \\ L \leftarrow L \setminus L_{\text{diff}} \end{cases}$

main
 $N \leftarrow \emptyset; \quad L \leftarrow \emptyset$
 extendSupportGraph$(O, \{C | C : c \in \mathbf{C}\} \cup \{O\})$
 prune()
 return $(\langle N, L\rangle)$

of its descendants is observed). Thirdly, the approach by Timmer et. al. returns a tree whereas Algorithm 4.1 returns a graph.

New nodes corresponding to variables or inferences are only created and added to the support graph if these do not already exist, so as to ensure that each node is unique. When a node is created, it is initialised with content that will inform the explanation. If required, getOrCreateVariable((V)) initialises a variable object with the following content:

type: variable
id: V
prior: $Pr(V\|\mathbf{C})$
posterior: $Pr(V\|O:o,\mathbf{C})$
prior_label: $l(Pr(V\|\mathbf{C}))$
posterior_label: $l(Pr(V\|O:o,\mathbf{C}))$
change: $c(Pr(V\|\mathbf{C}), Pr(V\|O:o,\mathbf{C}))$

where l is a function that translates a probability to a verbal description of that probability and c is a function that produces a verbal description of the change in probability from prior to posterior. For demonstration purposes, the scale proposed by Witteman and Renooij [25] (reproduced in Table 1a) is used as function l. As no standard verbal-numerical scales expressing change in probability are available, a simple scale expressing degrees of change is proposed in Table 1b. Obviously, this work does not depend on this scale and it could be substituted for another.

Inferences of the form $V_{\text{parent}} \dashrightarrow V_{\text{child}}$ are called "influences". If required, getOrCreateInfluence() $(V_{\text{parent}} \dashrightarrow V_{\text{child}})$ initialises an influence object with the following content:

type: influence
id: $V_{\text{parent}} \dashrightarrow V_{\text{child}}$
from: V_{parent}
to: V_{child}

Inferences of the form $V_1 \dashrightarrow V_{\text{child}} \dashleftarrow V_2$ are called "synergies". If required, getOrCreateSynergy($(V_1 \dashrightarrow V_{\text{child}} \dashleftarrow V_2)$) methods initialises a synergy object with the following content:

type: synergy
id: $V_1 \dashrightarrow V_{\text{child}} \dashleftarrow V_2$
from: V_1
to: V_2
common_child: V_{child}
observations: $\{V : v \in \mathbf{C} \cup \{O : o\} \mid V \in \text{Desc}(V_{\text{child}})\}$

The non-bold nodes and fields in Figure 5 shows the support graph that is produced by Algorithm 4.1. It consists of the two paths through which the observation $P : p$ affects the variable of interest G. The first path is $P \rightarrow [G \dashrightarrow P] \rightarrow G$, which is added through the addParent() procedure. The second path is $P \rightarrow [P \dashrightarrow C \dashleftarrow E] \rightarrow E \rightarrow [G \dashrightarrow E] \rightarrow G$, which is added through the addCoparent() procedure followed by addParent(). Note that there is no path $P \rightarrow [P \dashrightarrow C] \rightarrow C \rightarrow [E \dashrightarrow C] \rightarrow E$ in the support graph as the algorithm explicitly blocks paths from one parent P to a co-parent E via a shared child C.

4.2 Inference effect annotation

Using the support graph on its own, it is possible to produce texts that list the various ways in which an observation affects a variable of interest. While such texts may notionally perform the task of explaining a result, they are unlikely to form compelling explanations to the reader. To make an explanation more compelling, it ought to focus on the most significant effects. It should also compare and contract competing effects, to help make the reader not only aware of the variables and inferences that explain the outcome, but also of those that are most likely to change the outcome. To make this possible, the support graph is extended with an indication of the direction and magnitude of inferences.

Qualitative probabilistic network theory provides the basic constructs needed to compute the direction of inferences [23]. The direction of influence $V_{\text{parent}} \dashrightarrow V_{\text{child}}$ is said to be positive if, for any set $\mathbf{X}$ of variable-value assignments of parents of V_{child} other than V_{parent}, the following property holds:

$$Pr\left(v_{\text{child}} | v_{\text{parent}} \mathbf{X}\right) \geq Pr\left(v_{\text{child}} | \overline{v_{\text{parent}}} \mathbf{X}\right) \tag{1}$$

Similarly, zero/neutral and negative influences are defined by substituting $\geq$ in (1) by $=$ and $\leq$ respectively. Later developments in the field of qualitative probabilistic networks have sought to measure the magnitudes of influences [18, 21]. Typically, these approaches work by defining an lower bound l and an upper bound u as follows:

$$l \leq \left| Pr\left(v_{\text{child}} | v_{\text{parent}} \mathbf{X}\right) - Pr\left(v_{\text{child}} | \overline{v_{\text{parent}}} \mathbf{X}\right) \right| \leq u \tag{2}$$

for any set $\mathbf{X}$ of variable-value assignments of parents of V_{child} other than V_{parent}. In line with these semi-quantitative approaches to qualitative reasoning, a measure of the effect of an influence $V_{\text{parent}} \dashrightarrow V_{\text{child}}$ for a given set $\mathbf{X}$ of variable-value assignments of parents of V_{child} other than V_{parent} is given by:

$$\begin{aligned} \text{effect}\left(v_{\text{child}} | v_{\text{parent}}, \mathbf{X}\right) = & Pr\left(v_{\text{child}} | v_{\text{parent}}, \mathbf{X}\right) - \\ & Pr\left(v_{\text{child}} | \overline{v_{\text{parent}}}, \mathbf{X}\right) \end{aligned} \tag{3}$$

(3) yields a value in the range $[-1, 1]$. The sign of this effect value corresponds to the direction of effect. Its absolute value provides an indication of the magnitude of effect. For the purposes of generating an explanatory text, such a metric must be translated into a descriptive label. This requires a totally ordered set L and a function $l : [-1, 1] \mapsto L$ such that:

$$\forall x, y \in [-1, 1], x \leq y \rightarrow l(x) \leq l(y) \tag{4}$$

For demonstration purposes, this paper will be using the labelling function given in Table 1c.

It is important to note that the effect of an influence $V_{\text{parent}} \dashrightarrow V_{\text{child}}$ depends on the context specified in $\mathbf{X}$ and that some or all variables in $\mathbf{X}$ may not have been observed. The probability distribution of unobserved variables in $\mathbf{X}$ may change as a result of making an observation elsewhere in the the network. To capture this information in the support graph, influence type nodes are extended with a prior and posterior probability distribution of effect labels. A probability distribution of effects for a given set of observations $\mathbf{Y}$ equals:

$$p_{\mathbf{Y}} : L \mapsto [0, 1] : p(e) = \sum_{\forall \mathbf{X},\, e = l(\text{effect}(V_{\text{child}} | V_{\text{parent}} \mathbf{X}))} Pr(\mathbf{X}|\mathbf{Y}) \tag{5}$$

Each influence type node in the support graph is extended with a "prior_effects" field containing $p_{\mathbf{C}}$ and a "posterior_effects" field containing $p_{\mathbf{C} \cup \mathbf{O}:\mathbf{o}}$. In the example problem, there are only two simple influences $G \dashrightarrow P$ and $G \dashrightarrow E$, where neither P nor E have

$Pr(x)$	Label
1	Certain
[0.85, 1)	Almost certain
[0.75, 0.85)	Probable
(0.5, 0.75)	Expected
0.5	Fifty-fifty
[0.25, 0.5)	Uncertain
[0.15, 0.25)	Improbable
(0, 0.15)	Almost impossible
0	Impossible

(a) Verbal-numerical scale expressing probabilities [25]

$Pr(x\|C, o) - Pr(x\|C)$	Label
(0.3, 1]	considerable increase
(0.15, 0.3]	substantial increase
(0.05, 0.15]	moderate increase
(0.01, 0.05]	slight increase
(0, 0.01]	inconsequential increase
0	unchanged
[−0.01, 0)	inconsequential decrease
[−0.05, −0.01)	slight decrease
[−0.15. − 0.05)	moderate decrease
[−0.3, −0.15)	substantial decrease
[−1, −0.3)	considerable decrease

(b) Verbal-numerical scale expressing changes in probability

Label	Effect	Description
+4	(0.5, 1]	strongly positive effect
+3	(0.25, 0.5]	moderate positive effect
+2	(0.125, 0.25]	fair positive effect
+1	(0, 0.125]	slight positive effect
0	0	neutral
−1	[−0.125, 0)	slight negative effect'
−2	[−0.25, −0.125)	fair negative effect
−3	[−0.5, −0.25)	moderate negative effect
−4	[−1, −0.5)	strong negative effect

(c) Verbal-numerical scale expressing magnitudes of inference effect

Table 1: Qualitative labels for the magnitudes of probabilities, labels and inferences

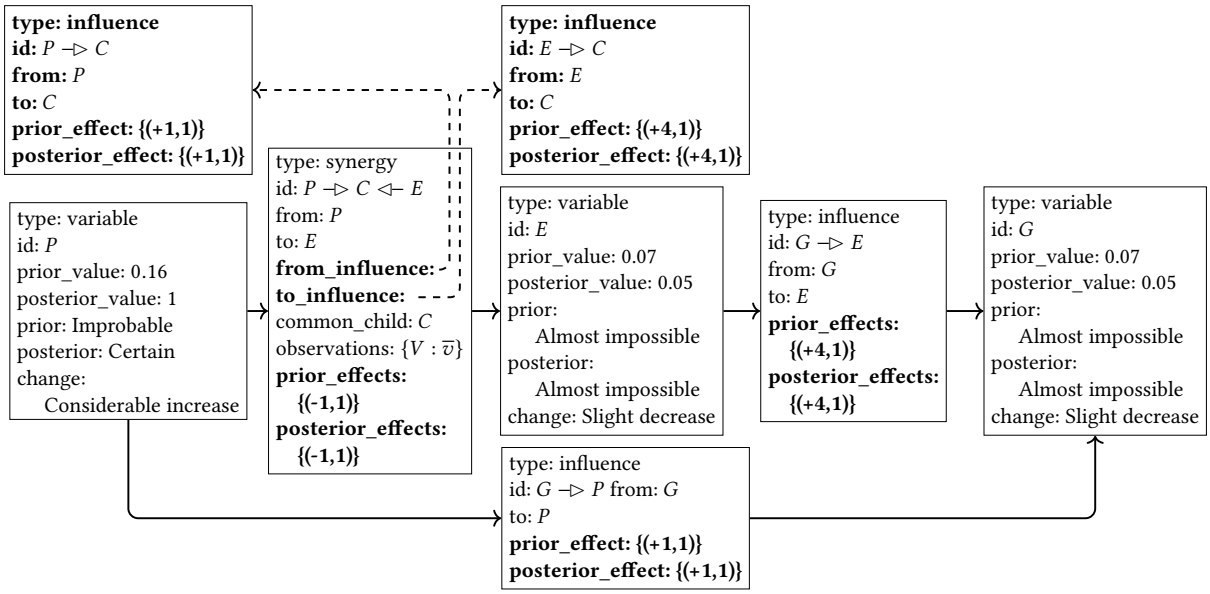


Figure 5: Support graph for the Jury observation fallacy problem after annotation with direction and magnitude of inferences. Fields and nodes containing fields shown in normal font are added in the initial construction phase discussed in Section 4.1. Fields and nodes containing only shown in bold font are added in the second phase discussed in Section 4.2.

parents other than G. Each influence has, therefore, only one effect. In this case:

$$\text{effect}(P|G) = Pr(p|g) - Pr(p|\overline{g}) = 0.1 - 0.0001 \approx 0.1$$
$$\text{effect}(E|G) = Pr(e|g) - Pr(e|\overline{g}) = 0.95 - 0.000001 \approx 0.95$$

Product synergies model the effects of inter-causal reasoning where there are converging connections $V_1 \dashrightarrow V_{\text{child}} \dashleftarrow V_2$ and V_{child} or one of its descendants has been observed [24]. Let c correspond to a possible value of V_{child}. There is a positive product synergy V_1 and V_2 with regards to value c of V_{child} if, for any set $\mathbf{X}$ of variable-value assignments of parents of V_{child} other than V_1 and V_2:

$$Pr(c|v_1v_2\mathbf{X})\,Pr(c|\overline{v_1}\,\overline{v_2}\mathbf{X}) \geq Pr(c|v_1\overline{v_2}\mathbf{X})\,Pr(c|\overline{v_1}v_2\mathbf{X}) \quad (6)$$

Zero/neutral and negative product synergies are in the same way defined by substituting $\geq$ in (6) by $=$ and $\leq$ respectively. Similar to influences, the effect of a synergy between V_1 and V_2 with regards to value c of V_{child} and a set $\mathbf{X}$ of variable-value assignments of parents of V_{child} other than V_1 and V_2 is given by:

$$\begin{aligned}\text{effect}(c|V_1, V_2, \mathbf{X}) = &Pr(c|v_1v_2\mathbf{X})\,Pr(c|\overline{v_1}\,\overline{v_2}\mathbf{X}) - \\ &Pr(c|v_1\overline{v_2}\mathbf{X})\,Pr(c|\overline{v_1}v_2\mathbf{X})\end{aligned} \quad (7)$$

Similar to (3), (7) yields a value in the range$[-1, 1]$, the sign of this effect value corresponds to the direction of effect and the value

provides an indication of the magnitude of effect. Product synergy effects are incorporated in the support graph in the same way as influence effects, using a totally ordered set L and an associated labelling function l that satisfies (4). For demonstration purposes, the labelling function of Table 1c will be used for product inferences as well.

A probability distribution of effects for a given set of observations $\mathbf{Y}$ equals:

$$p_{\mathbf{Y}} : L \mapsto [0,1] : p(e) = \sum_{\forall c \forall \mathbf{X}, e=l(\text{effect}(c|V_1,V_2,\mathbf{X}))} Pr(\mathbf{X}|\mathbf{Y}) \quad (8)$$

Each influence type node in the support graph is extended with a "prior_effects" field containing $p_{\mathbf{C}}$ and a "posterior_effects" field containing $p_{\mathbf{C} \cup \mathbf{O} \cdot \mathbf{o}}$. In the example problem, there is one product synergy with the following effects:

$$\begin{aligned}\text{effect}(c|P,E) &= Pr(c|p,e)\,Pr\left(c|\overline{p},\overline{e}\right) - Pr(c|p,\overline{e})\,Pr\left(c|\overline{p},e\right)\\ &= 0.9999999 \times 0.00001 - 0.99 \times 0.02 \approx -0.02\\ \text{effect}(\overline{c}|P,E) &= Pr(\overline{c}|p,e)\,Pr\left(\overline{c}|\overline{p},\overline{e}\right) - Pr(\overline{c}|p,\overline{e})\,Pr\left(\overline{c}|\overline{p},e\right)\\ &= 0.0000001 \times 0.99999 - 0.01 \times 0.98 \approx -0.01\end{aligned}$$

Given that $Pr(c|\overline{v}) = Pr(c|\overline{v},p) = 1$, the product synergy equals approximately -0.02 in all circumstances applicable to the query in the example scenario.

Useful explanations of synergistic effects via a head-to-head connection $V_1 \dashrightarrow V_{\text{child}} \dashleftarrow V_2$ will depend on the nature the constituent influences $V_1 \dashrightarrow V_{\text{child}}$ and $V_2 \dashrightarrow V_{\text{child}}$. For example, if the synergistic effect between V_1 and V_2 is consistently negative but both V_1 and V_2 have a consistently positive effect on V_{child}, the situation matches the classic "explaining away" scenario: both V_1 and V_2 are explanations for V_{child} and the presence of one explanation makes the need for an alternative explanation less pressing [24]. This situation occurs in $P \dashrightarrow C \dashleftarrow E$: intuitive, while it is unlikely for a defendant to be charged without there being hard evidence of the defendant's guilt, this is somewhat more likely when the defendant has committed priors offences.

To incorporate this information in the support graph, for any given synergy $V_1 \dashrightarrow V_{\text{child}} \dashleftarrow V_2$, corresponding influence nodes $V_1 \dashrightarrow V_{\text{child}}$ and $V_2 \dashrightarrow V_{\text{child}}$ are created in same way as described above. The synergy is extended with two fields named "from_influence" and "to_influence", and these are linked to the newly created influence nodes. In the Jury observation example support graph, two influence nodes are created corresponding to $P \dashrightarrow C$ and $E \dashrightarrow C$ and linked from their associated synergy fields.

The final support graph after extending it with direction and magnitude information about inferences and influences associated with synergies is presented in Figure 5. This model provides all the information needed to produce explanatory texts in a structured manner.

5 HIGH-LEVEL PLANNING

Section 4 has presented an approach to produce a content model from which a narrative is to be generated. Conventional approaches to explaining a BN or belief propagation within it tend to verbalise an entire model or process. The problems with such an approach is that it ignores the objective of the narrative.

What is the purpose of explaining the outcome of the query in the Jury observation fallacy model? Arguably, at the outset it is to explain how it was possible for the probability of guilt to go down as a consequence of observing that the defendant committed prior offences. In other words, the explanation needs to identify the mechanism by means of which it was possible for $Pr(g)$ to reduce as a typical reader, when first confronted with the model, will not have expected this to be possible. Once this mechanism has been identified, the reader may wish to understand how its effect is stronger than other inference chains. The latter explanation requires more depth and a focus on the strength of effects.

There are two predominant techniques to plan the content of a narrative: schema-based approaches and planning-driven approaches based on rhetorical structure theory [4]. Schema-based approaches are rooted in the notion that there are standard ways of communication certain ideas [16]. As such, these approaches employ formally defined schemata or templates that specify structures (sequences, iterations, etc.) of message-types that, in combination, achieve a particular goal. Their main limitation is that the schemata do not stipulate how the constituent messages contribute to the overall goal. Rhetorical structure theory addresses this by defining finer-grained schemata that stipulate how certain combinations of phrases contribute to rhetorical sub-goals [15]. This enables the use of planning techniques to find ways in which goals can be achieved by instantiating and applying schemata in a particular order [4].

An in-depth specification of a representation formalism and inference mechanism for this planning method is beyond the scope of this paper. Instead, this section elaborates how high-level planning can be applied in this context. In essence, a plan is a sequence of actions that bring a system from an initial state to a goal state. In the context of NLG, the system corresponds to the reader, the goal corresponds to enabling the reader to achieve a certain regard for a set of statements, and the actions correspond to communicating certain sentences or sequences of sentences.

A common approach to AI planning involves defining action templates, where the action template identifies an effect, some conditions or prerequisites for applying the action template, followed by one or a sequence of actions to be applied. For example, in the Jury observation fallacy problem, the communicative goal may be to increase the reader's regard for the possibility that the probability the defendant is guilty decreases if it is observed that the defendant has committed prior offences. This requires an explanation that there exists a path from P to G in the support graph whose combined effect is negative. This can be achieved by explaining that there exists one path that has a negative effect on the probability of the associated variable's truth. Action template 1 formalises this idea:

Action template 1. Explain the possibility of negative effect	
Effect	explain(effect_is_possible(V,negative))
Conditions	N.id=V
	V.change<unchanged
	X=path(O,V)
	effect_sign(X,negative)
Action	single_path_justifies_effect_is_possible
	explain(effect_sign(X,negative))

Variable	Parsed proposition
G	(S (NP (DT *the*) (NN *defendant*)) (VP (VBZ *is*) (ADJP (JJ *guilty*))))
P	(S (NP (DT *the*) (NN *defendant*)) (VP (VBD *committed*) (NP (JJ *prior*) (NNS *offences*))))
E	(S (NP (EX *there*)) (VP (VBZ *is*) (NP (NP (JJ *hard*) (NN *evidence*)) (VP (VBG *supporting*) (NP (DT *the*) (NN *defendant's*) (NN *guilt*))))))
C	(S (NP (DT *the*) (NN *defendant*)) (VP (VBZ *is*) (VP (VBN *charged*))))
V	(S (NP (DT *the*) (NN *defendant*)) (VP (VBZ *is*) (VP (VBN *found*) (S (ADJP (JJ *guilty*))))))

Table 2: The grammatical structure of the propositions in the Jury observation fallacy problem, as parsed by OpenNLP using the `en-parser-chunking` model.

In the support graph of Figure 5, there is one such path: $(P \rightarrow [P \dashv\triangleright C \triangleleft\!\!- E] \rightarrow \ldots \rightarrow G)$.

The end result of the high-level planning phase is a sequence of atomic actions (i.e. actions that cannot be decomposed further) that achieve the goal of the communication. For example, to explain that it is possible for observing p to cause a decrease of $Pr(g)$, the following plan could be produced:

- Action: Describe the approach to answering the question: that identifying the existence of a single path with a given effect justifies why that effect is possible.
- Sub-goal: Explain that the path $P \rightarrow [P \dashv\triangleright C \triangleleft\!\!- E] \rightarrow \ldots \rightarrow G$ causes the probability of $Pr(g)$ to decrease.
  - Action: State there is a path $P \rightarrow [P \dashv\triangleright C \triangleleft\!\!- E] \rightarrow \ldots \rightarrow G$ that causes the probability of $Pr(g)$ to decrease.
  - Sub-goal: Explain the sign of path $P \rightarrow [P \dashv\triangleright C \triangleleft\!\!- E] \rightarrow \ldots \rightarrow G$ is negative:
    - Action: Explain how the sign of path $P \rightarrow [P \dashv\triangleright C \triangleleft\!\!- E] \rightarrow E$ is negative.
    - Action: Conclude that this causes $Pr(e)$ to decrease.
    - Action: State that the decrease of $Pr(e)$ is propagated to G through positive influences.

6 MICRO-PLANNING

The atomic actions a high-level plan provide instructions to produce individual sentences that, in sequence, form an explanatory narrative. These sentences can be constructed by means of sentence templates that, like actions, prescribe how an atomic action can be completed under certain conditions by applying a given sentence template. The templates differ from actions in that they specify grammatical constructions that combine sentences and/or phrases identified in the conditions into more substantial constructs.

6.1 Variables

As noted in Section 2, it is assumed that all variables, except those observed in context **C**, are Boolean variables corresponding to propositions. To produce explanations, each variable must come with a sentence describing the proposition. For the purpose of using these sentences for generating natural language explanations, the grammatical structure of the sentences must specified with part-of-speech (POS) tags. In what follows, the POS tags of the Treebank II Style of Penn Treebank Project are used [6]. Table 2 lists the specifications of the propositions describing the variables in the Jury observation fallacy model of Figure 1. This Table specifies, for instance, that the description of G is "the defendant is guilty", where "the defendant" is a noun phrase (NP), "is guilty" is a verb phrase (VP), "guilty" is an an adjective phrase (ADJP) consisting only of the adjective (JJ) "guilty".

All information required to explain the state and changes of variables are incorporated in the support graph. But, these can be described in different ways, providing varying levels of detail, depending on the role of these variables in the explanation. The structure of an explanation and its intended narrative purpose is defined by means of Template 1.

Sentence template 1. Variable	
Effect	explain(V,change)
Conditions	node(N) N.name = V N.type = variable
Template	(S (NP *the probability*) (SBAR (IN *that*) (S V**.proposition**)) (VP (V?? N**.change**) (PP (IN *from*) (ADJP (JJ N**.prior**) (PP (TO *from*) (ADJP (JJ N**.posterior**)))))))

When applied to variable node P in the support graph, this template produces the explanation: "*the probability that* [the defendant committed prior offences] [to increase considerably] *from* [improbable] *to* [certain]". This can be realised as: "*The probability that the defendant committed prior offences has increased considerably from improbable to certain.*"

6.2 Influences

Producing explanations of influences is somewhat more challenging because the text needs to describe the relationship between two variables from sentences describing the propositions they represent. This problem is decomposed into smaller tasks. The task of describing the magnitude of effect of an influence between two variables, for example, can be decomposed into two parts: (i) generate an expression of the magnitude of effect and (ii) use that expression in a description of the relationship between the two propositions.

Sentence template 2. Consistently strong effect NP	
Result	effect_np(I)
Conditions	I.type ∈ {influence,synergy} $Pr(I.\text{prior\_effects} = +4) = 1$ $Pr(I.\text{posterior\_effects} = +4) = 1$
Template	(NP (DT *a*) (ADJP (RB *consistently*) (JJ *strong*)) (JJ *positive*) (NN *effect*))

Expressions of magnitude of effect are derived from the content model of Section 4, which provides prior and posterior distributions

of effect labels. These distribution must be described by means of a single statement, which can be more or less conclusive depending on the nature distributions of effect labels. For example, Template 2 is applicable to influence nodes $G \dashrightarrow E$ and $E \dashrightarrow C$ in of Figure 5.

The expressions of magnitude of effect, such as "a consistently strong positive effect" are used by another set of templates to generate descriptions of the effect of an inference between two propositions. One generic template that applies to any influence between a pair of variables that can each be expressed by a sentence consisting of a noun phrase followed by a verb phrase is Template 3.

Sentence template 3. Standard inference magnitude description	
Result	describe(I,effect_magnitude)
Conditions	I.type ∈ {influence,synergy} N_{from}.id = V_1.from n_{from}>Unchanged N_{to}.id = V_2.to N_{from}.proposition = (S (NP X_{from}) (VP Y_{from})) N_{to}.proposition = (S (NP X_{to}) (VP Y_{to})) (NP Z) = qualification(I,effect)
Template	(S (NP (NP *an increase*) (PP in the probability) (SBAR (IN *that*) (S (NP $X_{\textbf{from}}$) (VP $Y_{\textbf{from}}$)))) (VP (VBZ *has*) (NP (NP Z) (PP *on the probability*)) (SBAR (IN *that*) (S (NP $X_{\textbf{to}}$) (VP $Y_{\textbf{to}}$)))))

When applied to influence $G \dashrightarrow E$ this template produces the following explanation: "*An increase in the probability that the defendant is guilty has a consistently strongly positive effect on the probability that there is hard evidence supporting the defendant's guilt.*" Generic descriptions of influences, such as the example above, can unhelpful, especially if they are lengthy and clunky. More eloquent sentence structures can often be found by employing the structure of proposition descriptors. For example, it is common for a pair of propositions to share the same subject, as in $G \dashrightarrow P$. Template 4 defines how such an influence can be described.

Sentence template 4. Shared subject influence	
Result	describe(I,effect)
Conditions	I.type = influence N_{from}.id = V_1.from N_{to}.id = V_2.to N_{from}.proposition = (S (NP X) (VP ADJP Y_{from})) N_{to}.proposition = (S (NP X) (VP Y_{to})) (ADVP Z) = qualification(I,effect)
Template	(S (NP (plural X) (SBAR (WHNP (WP *who*)) (S (VP $Y_{\textbf{from}}$) (VP (VBP *are*) (ADVP Z (S (VP (TO *to*) (VP $Y_{\textbf{to}}$)))))))))

Template 4 requires that the direction and magnitude of an influence to be described as an adverb phrase rather than a noun phrase, which can be produced by Template 5:

Sentence template 5. Slightly positive effect advp	
Result	probability_relation_advp(I)
Conditions	I.type ∈ {influence,synergy} Pr(I.prior_effects = +1) = 1 Pr(I.posterior_effects = +1) = 1
Template	(ADVP (RB *slightly*) (RBR *more*) (RB *likely*))

This allows Template 4 to generate the description: "[plural the defendant] *who* [is guilty] are slightly more likely to [has committed prior offences]", which can be realised as: "*Defendants who are guilty, are slightly more likely to have committed prior offences.*" Observe how this is a much more articulate description of the influence the one based on the generate Template 3, which would be: "An increase in the probability that the defendant is guilty has a consistently slightly positive effect on the probability that the defendant has committed prior offences".

6.3 Synergies

Although synergistic relationships involve three variables via a head-to-head connection $V_1 \dashrightarrow V_{\text{child}} \dashleftarrow V_2$, they are in fact relationships between pairs of variables V_1 and V_2. Therefore, it is possible to describe the relationship between V_1 and V_2 in the same way as a conventional influence between V_1 and V_2. For example, using Template 3 the synergy $P \dashrightarrow C \dashleftarrow E$ can be described by: "*An increase in the probability that the defendant has committed prior offences has a consistently slight negative effect on the probability that there is hard evidence supporting the defendant's guilt.*" While this description characterises the effect of a change in belief that the defendant committed prior offences on the belief that there is hard evidence of the defendant's guilt, it does not explain what caused that effect. These synergistic effects can be particularly difficult to understand and a deeper explanation may be required. A deeper explanation of the synergistic effect would require a higher level plan that uses template instances to construct sentences. The next section uses this problem as an example to illustrate how the techniques presented herein can be combined to produce a natural language explanation.

7 EXAMPLE

To illustrate the overall approach, this section explains how the negative product synergy $P \dashrightarrow C \dashleftarrow E$ can be explained. The content required for the explanation is available in the associated synergy node N of the support graph of Figure 5. The following plan can be used to explain a negative product synergy where the constituent influences both have a positive effect:

(1) Describe the observations of N.
(2) Describe the consequence of this on the value/distribution of the N.common_child.
(3) State that there are two variables that help explain the state of N.common_child:
(4) Describe the proposition N.from, and
(5) Describe the proposition N.to
(6) State that either of these explanations makes the other less necessary to explain that the defendant is charged.
(7) Describe the synergy effect.

Some of the components of the plan have been defined by means of templates in Section 6. Without defining further templates, below is an example of how the above plan can be implemented by means of templates. In this plan, each of the lines would be produced by a template. To show how the text is constructed by templates, the propositions associated with variables are presented in italic and constituent template instantiations are underlined.

(1) *The defendant is found not guilty.*
(2) As a consequence of this, it is certain that *the defendant is charged.*
(3) There are two variables that help explain why *the defendant is charged* as the likelihood of this event increases with the probability that:
(4) *the defendant committed prior offences* and
(5) *there is hard evidence supporting the defendant's guilt.*
(6) Either of these explanations makes the other less necessary to explain that the defendant is charged.
(7) Therefore, an increase in the probability that *the defendant has committed prior offences* has a consistently slight negative effect on the probability that *there is hard evidence supporting the defendant's guilt.*

8 CONCLUSIONS AND FUTURE WORK

This paper has presented an approach to combine natural language generation (NLG) techniques [4] with the support graph approach introduced in [13] to produce natural language explanations of queries seeking to assess the effect of an observation of one variable in a Bayesian network (BN) on another variable, in the context of a set of prior observations.

The approach uses a support graph annotated with information concerning changes in probability distributions of variables in the network and information about the nature of dependencies between relevant connected nodes to produce a model that captures the information required to produce explanatory text. While this support graph provides a form of explanation in its own right (and was conceived for the purpose of producing explanatory arguments [22]), it still has certain limitations. Firstly, a support graph is a complete model that identifies all inferences that are potentially relevant. Complete explanations are not necessarily the best explanations as they lack focus. This paper has identified how high-level plan templates can help narrow down which parts of a support graph are relevant to provide certain information and how plans can help structure the explanatory narrative. Secondly, some of the information contained in a support graph can be difficult to understand. For example, synergistic head-to-head connections in a BN can introduce dependencies whose nature and importance human decision makers find hard to understand. This paper has identified how elements of the support graph can be converted into sentences using sentence-level templates. To construct these sentences, the approach requires minimal problem specific information: only one sentence must be produced per variable describing the proposition that variable represents.

The broader purpose of this paper is to identify a research agenda that aims to produce tools that generate better explanations of BNs and their application. The next step in this work is to implement the content model generation method and some of the NLG techniques presented herein, with a view to evaluate the approach on a range of different BNs and associated queries. The high-level planning and micro-planning components require significant further work. The language used for specifying actions and templates needs to be refined so that it can effectively represent results, preconditions and actions/templates. In the longer term, the work will be extended to BNs with non-Boolean variables and evaluated on a range of sample BNs.

REFERENCES

[1] C. Aitken and F. Taroni. 2004. *Statistics and the Evaluation of Evidence for Forensic Scientists* (2nd edition ed.). Wiley.
[2] C. Aitken, F. Taroni, and P. Garbolino. 2003. A graphical model for the evaluation of cross-transfer evidence in DNA profiles. *Theoretical Population Biology* 63 (2003), 179–190.
[3] S. Andersen, K. Olesen, F. Verner Jensen, and F. Jensen. 1989. HUGIN - A Shell for Building Bayesian Belief Universes for Expert Systems. *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence* 2, 1080–1085.
[4] J. Bateman and M. Zock. 2005. Natural language generation. In *The Oxford Handbook of Computational Linguistics*, R. Mitkov (Ed.). Oxford University Press, 284–305.
[5] A. Biedermann, F. Taroni, O. Delemont, C. Semadeni, and A. Davison. 2005. The evaluation of evidence in the forensic investigation of fire incidents. Part II. Practical examples of the use of Bayesian networks. *Forensic Science International* 147 (2005), 59–69.
[6] A. Bies, M. Ferguson, K. Katz, M. Macintyre, R.and Contributors, V. Tredinnick, G. Kim, M. Ann Marcinkiewicz, and B. Schasberger. 1995. Bracketing Guidelines For Treebank II Style Penn Treebank Project. (04 1995).
[7] R. Cook, I. Evett, G. Jackson, P. Jones, and J. Lambert. 1998. A Model for Case Assessment and Interpretation. *Science and Justice* 38, 6 (1998), 151–156.
[8] G. Davis. 2003. Bayesian reconstruction of traffic accidents. *Law, Probability and Risk* 2 (2003), 69–89.
[9] N. Fenton and M. Neil. 2013. *Risk Assessment and Decision Analysis with Bayesian Networks.* CRC Press.
[10] S. Gittelson, A. Biedermann, S. Bozza, and F. Taroni. 2013. Modeling the forensic two-trace problem with Bayesian networks. *Artificial Intelligence and Law* 21, 2 (2013), 221–252.
[11] J. Halpern. 2003. *Reasoning about uncertainty.* MIT Press.
[12] R. Hughes. 2017. Using a Bayesian Network to Predict L5/S1 Spinal Compression Force from Posture, Hand Load, Anthropometry, and Disc Injury Status. *Applied Bionics and Biomechanics* 2017 (2017), 2014961.
[13] J. Keppens. 2016. Explaining Bayesian Belief Revision for Legal Applications. In *Proceedings of the 29th International Conference on Legal Knowledge and Information Systems.* 63–72.
[14] C. Lacave and F. Díez. 2002. A review of explanation methods for Bayesian networks. *Knowledge Engineering Review* 17, 2 (2002), 107–127.
[15] W. Mann and S. Thompson. 1988. Rhetorical structure theory: towards a functional theory of text organisation. *Interdisciplinary Journal for the Study of Discourse* 8 (1988), 243–281. Issue 3.
[16] K. McKeown. 1985. Discourse strategies for generating natural-language text. *Artificial Intelligence* 27 (1985), 1–41.
[17] J. Mortera, A. Dawid, and S. Lauritzen. 2003. Probabilistic expert systems for DNA mixture profiling. *Theoretical Population Biology* 63 (2003), 191–205.
[18] S. Parsons. 1995. Refining Reasoning in Qualitative Probabilistic Networks. In *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence.* 427–434.
[19] J. Pearl. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference* (second edition ed.). Morgan-Kaufmann.
[20] R. Perera and P. Nand. 2017. Recent advances in natural language generation: a survey and classification of the empirical literature. *Computing and Informatics* 36 (2017), 1–32.
[21] S. Renooij, S. Parsons, and P. Pardieck. 2003. Using Kappas as Indicators of Strength in Qualitative Probabilistic Networks. In *Proceedings of the Seventh European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty.* 87–99.
[22] S. Timmer, J-J. Meyer, H. Prakken, S. Renooij, and B. Verheij. 2017. A two-phase method for extracting explanatory arguments from Bayesian networks. *International Journal of Approximate Reasoning* 80 (2017), 475–494.
[23] M. Wellman. 1990. Fundamental concepts of qualitative probabilistic networks. *Artificial Intelligence* 44 (1990), 257–303.
[24] M. Wellman and M. Henrion. 1993. Explaining "explaining away". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15 (1993), 287–291.
[25] C. Witteman and S. Renooij. 2003. Evaluation of a verbal-numerical scale. *International Journal of Approximate Reasoning* 33 (2003), 117–131.