Explanations by arbitrated argumentative dispute

Kristijonas Čyras a, b, David Birch a, Yike Guo a, Francesca Toni b, Rajvinder Dulay b, Sally Turvey b, Daniel Greenberg b, Tharindi Hapuarachchi b

a Department of Computing, Imperial College London, 180 Queen’s Gate, London SW7 2AZ, United Kingdom
b Thomson Reuters, 5 Canada Square, Canary Wharf, London E14 5AQ, United Kingdom

Abstract

Explaining outputs determined algorithmically by machines is one of the most pressing and studied problems in Artificial Intelligence (AI) nowadays, but the equally pressing problem of using AI to explain outputs determined by humans is less studied. In this paper we advance a novel methodology integrating case-based reasoning and computational argumentation from AI to explain outcomes, determined by humans or by machines, indifferently, for cases characterised by discrete (static) features and/or (dynamic) stages. At the heart of our methodology lies the concept of arbitrated argumentative disputes between two fictitious disputants arguing, respectively, for or against a case’s output in need of explanation, and where this case acts as an arbiter. Specifically, in explaining the outcome of a case in question, the disputants put forward as arguments relevant cases favouring their respective positions, with arguments/cases conflicting due to their features, stages and outcomes, and the applicability of arguments/cases arbitrated by the features and stages of the case in question. We in addition use arbitrated dispute trees to identify the excess features that help the winning disputant to win the dispute and thus complement the explanation.

We evaluate our novel methodology theoretically, proving desirable properties thereof, and empirically, in the context of primary legislation in the United Kingdom (UK), concerning the passage of Bills that may or may not become laws. High-level factors underpinning a Bill’s passage are its content-agnostic features such as type, number of sponsors, ballot order, as well as the UK Parliament’s rules of conduct. Given high numbers of proposed legislation (hundreds of Bills a year), it is hard even for legal experts to explain on a large scale why certain Bills pass or not. We show how our methodology can address this problem by automatically providing high-level explanations of why Bills pass or not, based on the given Bills and their content-agnostic features.

Keywords: Argumentation, Explanation, Legislative data

1. Introduction

Computational argumentation (as overviewed by Rahwan & Simari, 2009) is a branch of knowledge representation and reasoning within the field of Artificial Intelligence (AI) concerning reasoning with incomplete and conflicting information. In argumentation, particularly abstract argumentation (AA) (Dung, 1995), information is modelled via an argument graph (also commonly known as argumentation framework (AF)) whose nodes (arguments) represent pieces of information and (directed) edges represent conflicts between them. In the last decade argumentation has gained traction as a formalism to provide graph-based formal explanations in AI, to explain outputs determined algorithmically by machines. This is one of the most pressing and studied problems in Artificial Intelligence (AI) nowadays, e.g. as indicated in (Gunning, 2017; Holzinger, Kieseberg, Weiippl, & Tjoa, 2018; Zhang & Zhu, 2018). Argumentation has been used for explanation of e.g. classifications (Amgoud & Serrurier, 2008; Cocarascu, Čyras, & Toni, 2018), decisions (Amgoud & Prade, 2009), product recommendations (Briquenz et al., 2014; Rago, Cocarascu, & Toni, 2018), as well as argument (non-)acceptance (Fan & Toni, 2015b; García, Cheslevar, Rotstein, & Simari, 2013) in general. At a high-level, argument graphs model information and underlying conflicts in a manner that affords an argumentative interpretation by way of disputes (exchanges of conflicting arguments) between two parties. If the instantiated arguments carry concrete information such as ‘yes’ and ‘no’ decisions, the disputes may allow to explain the final decision by contrasting other relevant arguments.

https://doi.org/10.1016/j.eswa.2019.03.012
0957-4174/© 2019 Elsevier Ltd. All rights reserved.
In this paper we advance a novel argumentation-based methodology for explanation, focusing on the less studied but equally pressing problem of using AI to explain outputs determined by humans. Our methodology is somewhat inspired by case-based reasoning (CBR). CBR (see de Mántaras, 2001; Richter & Weber, 2013 for overviews) is widely used in AI in support of several applications. One distinguishing characteristic of CBR formalisms is their representation of information through cases. At high level, a case consists of a set of features (e.g. attribute-value pairs Sørmo, Cassens, & Aamodt, 2005) and an outcome (representing e.g. a ‘yes’/‘no’ decision). In our setting and methodology, features are accompanied by stages indicating the progression of cases, thus incorporating dynamic information and distinguished from features that are static.

By and large, CBR formalisms are concerned with adapting the outcomes of past cases (with known features and outcomes) to determine the outcomes of new cases (with known features but unknown outcomes). Instead, our methodology is concerned with explaining the outcomes of focus cases (with known features, stages and outcomes) by making use of the known features, stages and outcomes of given cases.

The explanations underpinning our methodology are obtained from suitably defined, instantiated argumentation frameworks (AFs) mined from the focus and given cases ad hand. In particular, arguments in these AFs are cases and the explanations for outcomes of focus cases are afforded by the argumentative reading of cases through arbitrated argumentative dispute. We realise this reading via arbitrated dispute trees, which are subgraphs of the AFs representing how two players exchange arguments (given cases) in favour of different outcomes and a given focus case arbitrates away the irrelevant arguments. We propose a way to identify in the arbitrated dispute trees the excess features that help the winning player to win the dispute. Such features together with the arbitrated dispute trees comprise explanations of case outcomes.

We study theoretical properties of our instantiated AFs and establish desirable characteristics of explanations. To illustrate our approach in a real-world setting, we apply our novel methodology to the UK Parliament legislative data whereby we aim at explaining why proposed legislation has or has not become law. To this end, we treat Parliamentary Bills as cases with content-agnostic features and stages and with outcomes of passing or not as laws. We use the arbitrated argumentative disputes and the excess features drawn from them to explain Bill outcomes. We do this automatically on a large (from a human perspective) corpus of Parliament data and evaluate our findings against legislative expert opinion.

The paper is structured as follows. We provide the necessary background on AA in Section 2. We describe our methodology in Section 3. In Section 4, we present a case study of applying our methodology to Parliamentary Bills. We review related work in Section 5 and conclude in Section 6.

2. Background

We here give the necessary background on abstract argumentation (AA). Throughout the paper, we use the terms argumentation framework (AF) and argument graph interchangeably.

Following Dung (1995), an AF is a pair (Args, ∼), where Args is a set of elements called arguments and ∼ is a binary relation on Args called the attack relation. For a, b ∈ Args if a ∼ b, then we say that a attacks b and that a is an attacker of b. If for a ∈ Args there is no b ∈ Args with b ∼ a, then a is unattacked. For a set of arguments E ⊆ Args and an argument a ∈ Args, E defends a if for all b ∼ a there exists c ∈ E such that c ∼ b.

In general, the set of arguments Args in an AF (Args, ∼) may be infinite, and ∼ may consist of infinitely many pairs. If both Args and ∼ are finite, then (Args, ∼) is said to be finite.

An AF provides means to represent conflicting information. Reasoning with that information is done by means of argumentation semantics. A semantics provides a characterisation of acceptable arguments in an AF. A set of acceptable arguments according to a semantics is called an extension and is taken as a reasoning outcome. Many semantics have been proposed, see e.g. (Baroni & Giacomin, 2009) for overviews. In this work we will consider the very well established grounded semantics: the grounded extension of (Args, ∼) can be constructed as G = ⋃i∈Ν Gi, where G0 is the set of all unattacked arguments, and ∀i ≥ 0, Gi+1 is the set of all arguments that Gi defends. For any (Args, ∼), the grounded extension G ∞ always exists and is unique. Moreover, the grounded extension can be easily computed bottom-up e.g. as in (Modgil & Caminada, 2009).

An AF (Args, ∼) can be seen as a directed graph with nodes Args and directed vertices ∼. In this work we will be particularly interested in acyclic AFs, whose directed graph is acyclic: a directed acyclic graph is a directed graph (N, V) without directed cycles, i.e. where no node is reachable from itself via a directed path containing at least one vertex; formally, ∃a0 ∈ N with either (d0, a0) ∈ V, or ∀a0 ≥ 1 such that (d0, a1), . . . , (an−1, an), (an, a0) ∈ V and ai ∈ N, ∀i ∈ {0, . . . , n}. The grounded extension of an acyclic AF is guaranteed to be non-empty and coincide with extensions under other argumentation semantics. In particular, it is guaranteed to be stable: for any (Args, ∼), E ∈ Args is a stable extension (Dung, 1995) iff E does not attack any argument in E and E attacks every argument a ∈ Args \ E.

3. Methodology

Our methodology relies upon drawing explanations for focus cases of interest, in need of explaining, and using given cases to obtain these explanations. Intuitively, a case amounts to a set of features, a sequence of stages it has been through, and an outcome at the latest stage. The choice of features, stages and outcomes is context dependent, as illustrated in Section 4. In this section we define our methodology abstractly, for generic choices.

Formally, we consider a fixed but otherwise arbitrary (possibly infinite) set Σ of features, and a finite sequence S = ⟨s1, . . . , sn⟩ with n ≥ 1, of stages. We let (T) denote the empty sequence and otherwise focus on subsequences of Σ of the form ⟨s1, . . . , sm⟩ for m ≤ n. We define a binary initial subsequence relation ⊆ over subsequences of Σ such that for S, S′ ∈ Σ, S ⊆ S′ iff either S′ = ⟨s1, . . . , si⟩, S = ⟨s1, . . . , si⟩ and k ≤ m ≤ n, or S′ = ⟨⟩. Further, we make use of the binary proper initial subsequence relation ⊂ defined by S′ ⊂ S iff S′ ⊆ S and S = S′. We also consider a set Ω = {δ, ˇδ} of two (distinct) outcomes, with δ called the default outcome (the choice of the default outcome is context dependent, as illustrated in Section 4). Then, the following defines cases.

Definition 3.1. Let F, S = ⟨s1, . . . , sn⟩ with n ≥ 1, and ∅ = {δ, ˇδ} be given.

- A case C is a triple C = (F, S, o) with θ ∈ F and θ ∉ S = ⟨s1, . . . , sn⟩ \ S, for some 1 ≤ m ≤ n, and o ∈ Ω.
- A case base is a finite set CB of cases that is coherent, i.e. for (F, S, o), (F′, S′, o′) ∈ CB, if F = F′ and S = S′, then o = o′.
- A focus case is a case, often denoted by φ = (Fφ, Sφ, oφ).

The distinction between features and stages in a case can be understood as follows. The features represent the information that is known about the case at any given stage. The stages represent how the cases progress through some pre-determined timeline. So the latest stage that a case ‘is in’ indicates how much that case has advanced in the order of stages. By imposing that the second component of a case is an initial subsequence of Σ we particularly impose the requirement that to reach a stage sm for m ∈ {1, . . . , n}, a case must pass through all stages s1, . . . , sm−1.
For an intuitive reading of cases, consider legislation. A case can represent a piece of proposed legislation that progresses through the legislature until either accepted as law or otherwise rejected. In addition to the actual contents, the proposed legislation carries content-agnostic features such as type of legislation, originating institution, sponsorship etc. Every piece of proposed legislation may carry different features and may also acquire new features, such as the type of committee that scrutinises the proposal, while progressing through the stages as determined by the legislative procedures. It is not unusual that all the proposed pieces of legislation flow through stages non-pre-emptively. A case base then represents a collection of accepted and rejected legislation, with various features and at various stages, where one may wish to focus on a piece of proposed legislation for inspection purposes. We will see concrete examples in the case study described in Section 4.

Note that a focus case is simply a case, which may in particular be extracted from the same repository of cases as CB. The difference between the focus case and the cases in CB is thus semantic rather than syntactic: the outcome of a focus case is in need of explanation, and the cases in CB may be used to provide this explanation.

To illustrate Definition 3.1, we use the following simple abstract example, used also throughout this section to illustrate several aspects of our methodology.

Example 3.2. Let $F = \{a, b, c\}$, $S = \{(s_1, s_2, s_3)\}$ and $\emptyset = \{+, -\}$ with -the default. Consider the following cases:

- $C_1 = \{(a), (s_1), +\}$, $C_6 = \{(a, b), (s_1), -\}$
- $C_2 = \{(b), (s_1, s_2), +\}$, $C_7 = \{(a, s_1, s_2, s_3), -\}$
- $C_3 = \{(b), (s_1), -\}$, $C_8 = \{(a, b), (s_1, s_2), +\}$
- $C_4 = \{(a, s_1), (s_1), -\}$, $\phi_1 = \{(a, c), (s_1, s_2), -\}$
- $C_5 = \{(a), (s_1, s_2), -\}$, $\phi_2 = \{(a, b, c), (s_1, s_2), +\}$

For intuition, think of the cases as legislative pieces as follows. There are three stages, $s_1, s_2$, and $s_3$, that each piece of legislation can go through. There are two outcomes, + and −, indicating that the proposed piece of legislation is respectively likely or unlikely to become a law. Feature a represents whether the legislation is national or international (think EU regulations); b represents whether the legislation is financial or not; c indicates public sponsorship.

Cases $C_1$ and $C_2$ have different features and different outcomes at the same stage $s_1$. Case $C_5$ has the same features as $C_1$, but $C_3$ has advanced further and has a different outcome. On the other hand, case $C_7$ with the same features as $C_5$ has advanced even further, but has the same outcome. Case $\phi_1$ has the additional feature c compared with $C_5$, and they are at the same stage and have the same outcome. Case $C_8$, on the other hand, has the additional feature b compared with $C_3$ and they are at the same stage, but have different outcomes.

Given this repository of cases, different choices of case base $CB$ and focus case are possible. In later illustrations, we will consider the specific choices of $CB = \{C_1, \ldots, C_8\}$ and focus case either $\phi_1$ or $\phi_2$. In general, the choice of $CB$ is context dependent, as we will see in Section 4, and the choice of focus case is dictated by what needs explaining.

We will relate the problem of explaining the outcome of a focus case to the membership problem within the grounded extension of a specific abstract argumentation framework (AF) (Dung, 1995) obtained from the case base, the focus case and the default outcome. We define the mapping onto this AF in two steps. The first step amounts to taking into account the case base and the default outcome alone, ignoring the focus case, whereas the second step takes the latter also into account to generate an AF from which explanations are drawn. In the reminder of this section we define the two steps and the explanations in turn.

3.1. AF corresponding to case base and default outcome

The first step in our methodology amounts to treating given cases and the default outcome as arguments and identifying conflicts between them, as follows.

Definition 3.3. The AF corresponding to a case base $CB$ and a default outcome $\delta \in \emptyset$, denoted by $AF(CB, \delta)$, is $\langle \text{Args}, \neg\rangle$ satisfying the following conditions

- $\text{Args} = CB \cup \{(\varnothing, \emptyset, \delta)\}$, where $(\varnothing, \emptyset, \delta)$ is called the default argument;
- for $(F, S, o), (F', S', o') \in \text{Args}$, it holds that $(F, S, o) \rightsquigarrow (F', S', o')$ if either
  1. $o \neq o'$, and $F \subseteq F'$ (different outcomes)
  2. $F = F'$ and $S \subseteq S'$ (specifiity)
  3. $F = F'$ and $S \subseteq S'$ (concision)
- or $F' = F$ and $S' \subseteq S$ (advancement)
- or $F = F'$ and $S \subseteq S'$ (proximity)

The default argument represents a presupposed bias towards the default outcome, and the default outcome can be interpreted as the outcome expected in the absence of information on features and stages.

For intuition in the setting of legislation, the default case can represent what is normally expected to happen to proposed legislation, depending on the legislation and the rules of conduct of legislation. For instance, until a piece of legislation is formally introduced and starts its journey in a particular part of legislature, it is not expected to become a law (think e.g. of a white paper).

In what follows, we refer to arguments and cases interchangeably (so, in particular, the default argument may be referred to as the ‘default case’). Also, unless specified otherwise, $(\text{Args}, \rightsquigarrow) = AF(CB, \delta)$ is the AF corresponding to $CB$ and $\delta$. Note that $(\text{Args}, \rightsquigarrow)$ is finite, as $CB$ is.

Some comments on Definition 3.3 are in order. The construction of $(\text{Args}, \rightsquigarrow)$ singles out conflicts between cases. In words, case $(F, S, o)$ attacks case $(F', S', o')$ whenever (i) they have different outcomes, and (ii) either a) $(F, S, o)$ is more specific and b) most concise such case (prioritised by features and then stages); (iii) or they have the same features but a) $(F, S, o)$ has advanced further and b) is the closest such case to $(F', S', o')$. The intuition for (ii) is as follows. If $(F, S, o)$ is more specific than $(F', S', o')$ and they have different outcomes, then $(F, S, o)$ can be seen as an exception to $(F', S', o')$ (at any stage). Being an exception means that the features in $F \neq F'$ sanction a change in the outcome from $o'$ to $o$, as standard in case-based reasoning. Concision ensures that only a minimal set of differentiating features causing that change is considered to sanction an exception. If multiple such sets exist, we prefer the closest one in terms of stages to identify as soon as possible when the change happens.

On the other hand, for (iii), if $(F, S, o)$ has advanced further than $(F', S', o')$ and flipped the outcome, then it can be understood that, for instance, some features are missing from the representation of $(F', S', o')$. That is, we may expect the latter case to acquire additional features as it passes through stages. Proximity then allows to identify the earliest stage when the change in the outcome occurs and one can expect additional features.

We illustrate as follows.

Example 3.4. $AF(CB, \neg)$ corresponding to $CB = \{C_1, \ldots, C_8\}$ and the default - from Example 3.2 is graphically depicted in Fig. 1. There and henceforth, nodes hold arguments/cases, and directed edges
represent attacks. We also name the default case as $C_0$, for ease of reference.

Note that $C_6$ does not attack $C_3$ while $C_2$ does, because $C_2$ has
(setwise) strictly more features than $C_3$. This illustrates the ‘specificity’ and
‘concision’ conditions in point (ii) of Definition 3.3. On the other hand, $C_7$ does not attack $C_1$ while $C_5$ does, because $C_7$ has
advanced further than $C_5$. This illustrates the ‘advance’ and
‘proximity’ conditions in point (iii) of Definition 3.3.

We note the following property of $AF(CB, \delta)$:

**Proposition 3.1.** $AF(CB, \delta)$ is a directed acyclic graph.

**Proof.** Clearly, $AF(CB, \delta) = (Args, \leadsto)$ is a directed graph. Suppose for a contradiction $(Args, \leadsto)$ has a directed cycle. As no case can
attack itself due to the different outcomes requirement for $\leadsto$, we find
$(F_0, S_0, o_0) \leadsto (F_1, S_1, o_1) \leadsto \ldots \leadsto (F_m, S_m, o_m) \leadsto (F_0, S_0, o_0)$ for some $m \geq 1$. It cannot be that any of the inclusions in $F_0 \subseteq F_1 \subseteq \ldots \subseteq F_m \subseteq F_0$ is proper, for otherwise $F_0 \not\subseteq F_0$. So $F_0 = F_0 \forall i$. But then $S_0 \subseteq S_1 \subseteq \ldots \subseteq S_m \subseteq S_0$, which implies $S_0 \subseteq S_0$. This is a contradiction. Hence, $(Args, \leadsto)$ has no directed cycles and is thus a directed acyclic graph. □

To illustrate, $AF(CB, \delta)$ from Example 3.4 is acyclic.

### 3.2. AF corresponding to case base, default outcome and focus case

In general, $AF(CB, \delta)$ affords a model of given cases and relationships among them. The second step in our methodology amounts to incorporating a given focus case into the model of given cases. Specifically, given a focus case of interest, we use the focus case to arbitrate which given cases are relevant towards explaining the outcome of the focus case. We do this by adding the focus case to $AF(CB, \delta)$ as a special argument that attacks all cases which either contain additional features (i.e. features not in the set of features of the focus case), or have advanced further than the focus case itself. Formally:

**Definition 3.5.** The $AF$ corresponding to a case base $CB$, a default outcome $\delta \in \emptyset$ and a focus case $\phi = (F_\phi, S_\phi, o_\phi)$, denoted by $AF(CB, \delta, \phi)$, is $(Args_\phi, \leadsto_\phi)$ satisfying the following conditions:

- $Args_\phi = Args \cup \{(F_\phi, S_\phi, o_\phi), (F, S, o) \in Args$ and either $F \not\subseteq F_\phi$ or $S \not\subseteq S_\phi\}$

Henceforth, unless specified otherwise, $(Args_\phi, \leadsto_\phi) = AF(CB, \delta, \phi)$ is the $AF$ corresponding to $CB, \delta$ and $(F_\phi, S_\phi, o_\phi)$.

**Example 3.6.** Recall the focus cases from Example 3.2:

$\phi_1 = ([a, c], (s_1, s_2), -)$ \quad $\phi_2 = ([a, b, c], (s_1, s_2), +)$

Let $AF(CB, \leadsto_\phi) = (Args_\phi, \leadsto_\phi)$ be the $AF$ corresponding to $CB$ and default - from Example 3.2 and focus case $\phi_i$, for $i = 1, 2$. For instance, $AF(CB, \leadsto_\phi_1)$ results from adding $\phi_1$ to $(Args, \leadsto)$ from Example 3.4 as in Definition 3.5, and is graphically depicted in Fig. 2 (the attacks from $\phi_1$ are slightly transparent to distinguish them from the others). Similarly, $AF(CB, \leadsto_\phi_2)$ results from adding $\phi_2$ to $(Args, \leadsto)$, and is graphically depicted in Fig. 3.

Clearly, since $AF(CB, \delta)$ is a directed acyclic graph, so is $AF(CB, \delta, \phi)$:

**Proposition 3.2.** $AF(CB, \delta, \phi)$ is a directed acyclic graph.

**Proof.** By Proposition 3.1, $AF(CB, \delta) = (Args, \leadsto)$ is a directed acyclic graph. Note that $(F_\phi, S_\phi, o_\phi)$ is not attacked in $(Args_\phi, \leadsto_\phi) = AF(CB, \delta, \phi)$, so its addition to $(Args, \leadsto)$ cannot introduce directed cycles. Since $(Args_\phi, \leadsto_\phi)$ has no other arguments that are not in $(Args, \leadsto)$, it is a directed acyclic graph too. □

For illustration, $AF(CB, \leadsto_\phi_1)$ and $AF(CB, \leadsto_\phi_2)$ from Example 3.6 are acyclic.
3.3. Explanations

The construction of \((\text{Args}_\delta \rightarrow \phi)\) completes the process of obtaining AFs to be used for generating explanations for outcomes of focus cases. We now proceed to formally define these explanations.

Conceptually, the basis of our explanations for the outcome of a focus case \((F_{\phi}, S_{\phi}, o_{\phi})\) can be seen as a dispute between two fictional players \(W\) (winner) and \(L\) (loser), arbitrated by the use of \((\theta, (\{\}, \delta))\) and \((F_{\phi}, S_{\phi}, o_{\phi})\). The root of the dispute is the default argument \((\theta, (\{\}, \delta))\). Depending on whether \(o_{\phi}\), \(\delta\) or \(\bar{\delta}\), the winner \(W\) aims to defend or attack the root \((\theta, (\{\}, \delta))\), and vice versa for the loser \(L\). Specifically, the winner \(W\) argues for the actual outcome \(o_{\phi}\) of the focus case using given cases whose outcomes coincide with \(o_{\phi}\); the loser \(L\) argues for the opposite outcome using given cases whose outcomes differ from \(o_{\phi}\). The two players put forward their arguments attacking each other, as they appear in \((\text{Args}_\delta \rightarrow \phi)\), starting with \((\theta, (\{\}, \delta))\). Moving arguments thus expresses the changes in the outcome due to exceptions or advancement through stages. The players \(W\) and \(L\) continue exchanging arguments from \((\text{Args}_\delta \rightarrow \phi)\) in this fashion until \((F_{\phi}, S_{\phi}, o_{\phi})\) arbitrates away the irrelevant arguments (by attacking them) and the dispute ends.

We formalise the process described above using arbitrated dispute trees, defined as follows.

**Definition 3.7.** Let \(\text{AF}(CB, \delta, \phi) = (\text{Args}_\delta \rightarrow \phi)\). An arbitrated dispute tree is a tree \(\mathcal{T}\) such that:
1. every node of \(\mathcal{T}\) is of the form \([x: \text{Args}_\delta]\) for \(x \in \{W, L\}\) and \(x \in \text{Args}_\delta\);
2. the root of \(\mathcal{T}\) is labelled by argument \((\theta, (\{\}, \delta))\) and is
   - a \(W\)-node, if \(o_{\phi} = \delta\); 
   - an \(L\)-node, else, if \(o_{\phi} = \bar{\delta}\);
3. for every \(W\)-node \(n\) labelled by some \(b \in \text{Args}_\delta\), and for every \(c \in \text{Args}_\delta\) such that \(c \rightarrow_{\phi} b\), there exists a child of \(n\), which is an \(L\)-node labelled by \(c\);
4. for every \(L\)-node \(n\) labelled by some \(b \in \text{Args}_\delta\), there exists exactly one child of \(n\) which is a \(W\)-node labelled by some \(c \in \text{Args}_\delta\) such that \(c \rightarrow_{\phi} b\);
5. there are no other nodes in \(\mathcal{T}\) except those given by 1–4.

The **defence set** of \(\mathcal{T}\), denoted by \(D(\mathcal{T})\), is the set of all arguments labelling \(W\)-nodes in \(\mathcal{T}\).

Intuitively, an arbitrated dispute tree can be seen to represent a debate—an exchange of related and relevant arguments—about an issue in question. For example, in discussing a certain piece of proposed legislation, the debate would amount to invoking other available related (in terms of features) and relevant (in terms of stages) legislation to explain why the outcome of the proposal in focus is as expected (default) or otherwise. More abstractly and generally, the suitability of argumentative trees for explanatory purposes has been well-discussed in the literature, which we will overview in Section 5.

**Example 3.8.** An arbitrated dispute tree \(\mathcal{T}_1\) with respect to \(\text{AF}(CB, \rightarrow, \phi_1)\) from Fig. 2 in Example 3.6 is depicted in Fig. 4. \(\mathcal{T}_1\) can be seen to represent an arbitrated dispute between \(W\) and \(L\) exchanging arguments supporting outcome \(\rightarrow\) of \(\phi_1\). Intuitively, to explain that the outcome of \(\phi_1\) is the default - in this case, the winner \(W\) aims to defend the default case \((\theta, (\{\}, \delta))\) by counter-attacking all the arguments moved by the loser \(L\). The disagreeing arguments are arbitratted away by the focus case \((F_{\phi}, S_{\phi}, o_{\phi})\), which can also be seen to “side” with \(W\). Ultimately, \(W\) succeeds as they have the last word, i.e. the leaves of \(\mathcal{T}_1\) are \(W\)-nodes.

There is another arbitrated dispute tree with respect to \(\text{AF}(CB, \rightarrow, \phi_1)\), namely \(\mathcal{T}_1^\prime\) as depicted in Fig. 5. The possible multitude of arbitrated dispute trees reflects the winner \(W\) having a multitude of cases in their favour with which to win the dispute.

For an arbitrated dispute tree with \(L\)-root, consider \(\mathcal{T}_2\), as depicted in Fig. 6, with respect to \(\text{AF}(CB, \rightarrow, \phi_2)\) in Fig. 3. Note that...
Fig. 6. An arbitrated dispute tree $T_2$ with respect to $AF(CB, -, \phi_2)$ from Example 3.6.

the focus case $\phi_2$ is not involved in this dispute, yet $\psi$ still success-
fully argues against the default case.
In general, both the focus case $(F_{\phi}, S_{\phi}, o_{\phi})$ and the default case $(\theta, \langle \cdot \rangle, \delta)$ act as arbiters in the dispute between $w$ and $l$ regarding the outcome $o_{\phi}$. To begin with, the arguments and their relationships are known and laid out in $AF(CB, \delta, \phi)$ and the arbiters need to decide how to best explain the outcome $o_{\phi}$. The explana-
tion starts with assigning the roles of the winner $w$ and loser $l$ with respect to the default: if default is the outcome of the focus case, then $w$ will be arguing for it; else, if default is not the outcome of the focus case, then $l$ will be arguing for the default instead, and $w$ against it. At the same time, $(F_{\phi}, S_{\phi}, o_{\phi}$) also arbi-
trates away the cases that are irrelevant to its outcome, if any.
Arbitrated dispute trees satisfy several properties irrespective of
their root label, as follows.

**Proposition 3.3.** Let $T$ be an arbitrated dispute tree (with the root
being either a $w$-node or an $L$-node), and $D(T)$ be its defence set (the
set of all arguments labelling $w$-nodes). Then the following hold.

(i) All leaves of $T$ are $w$-nodes.

(ii) If $(F_{\phi}, S_{\phi}, o_{\phi})$ appears in $T$, then it labels $w$-node(s) only.

(iii) $D(T) \subseteq G$, where $G$ is the grounded extension of $AF(CB, \delta, \phi)$.

**Proof.**

(i) By Definition 3.7 condition 4., every $L$-node has a child $w$-
node. Thus, every leaf of $T$ must be a $w$-node.

(ii) $(F_{\phi}, S_{\phi}, o_{\phi})$ is unattacked in $(Arg_{\phi}, \neg \neg \phi) = AF(CB, \delta, \phi)$, so
if it appears in $T$, it must be a leaf. By 1. above, $(F_{\phi}, S_{\phi}, o_{\phi})$
can thus label only a $w$-node.

(iii) Let $b \in D(T)$ and assume first that $\langle w : b \rangle$ is a leaf. Then by
Definition 3.7 condition 3., there is no $c \in Arg_{\phi}$ such that
$c \neg \neg \neg \phi b$. Hence, $b$ is unattacked in $(Arg_{\phi}, \neg \neg \phi)$. Thus, $b \in G$. Now, if $b$ does not label a leaf of $T$, then every one of its
attackers $c$ is labelling an $L$-node, and each such $c$ is in turn
necessarily attacked by an argument $a$ labelling a $w$-
node (Definition 3.7 condition 4.) The $a$ is either carried by
a $w$-leaf, or we iteratively repeat the same line of reasoning
until we hit a $w$-leaf. $D(T)$ thus defends every such $a$ and
hence defends $b$ too. Consequently, $b \in G$, and we ob-
in $D(T) \subseteq G$. □

This proposition essentially says that as a winner of the dispute,
player $w$ (i) has the last word, (ii) sides with the arbiter $(F_{\phi}, S_{\phi},
o_{\phi})$ (if the latter needs to arbitrate away any cases), and (iii) has
its arguments well-defended.

We believe these are desirable properties of arbitrated dispute
trees that intuitively make them suitable as explanations for why
the root argument is good. In particular, it seems almost neces-
sary that the winner of a dispute should have their arguments well-
defended, i.e. to have arguments against any counter-argument.
This implies that the arbiter should not be against the winner. As
a consequence, the last word, i.e. the unattacked arguments, should be
the winner's.

**Example 3.9.** In Example 3.8, all leaves of both arbitrated dispute
trees $T_1$ and $T_2$ are clearly $w$-nodes. The focus case $\phi_1$ ap-
pears in $T_1$ only, and it indeed labels only a $w$-node. Furthermore,
the grounded extensions of $AF(CB, -, \phi_1)$ and $AF(CB, -, \phi_2)$ are
$G_1 = \{\phi_1, C_4, C_5, C_6\}$ and $G_2 = \{\phi_2, C_4, C_5, C_6, C_7\}$, respectively, and we
clearly have $D(T_1) = \{\phi_1, C_4, C_6\} \subseteq G_1$ as well as $D(T_2) = \{C_8, C_4\} \subseteq G_2$.

The grounded extension $G$ of $AF(CB, \delta, \phi)$ is crucial when
determining whether and when an arbitrated dispute tree exists,
in the following sense.

**Proposition 3.4.** Let $G$ be the grounded extension of $AF(CB, \delta, \phi)$.

1. If $(\theta, \langle \cdot \rangle, \delta) \in G$ and $o_{\phi} = \delta$, then there exists an arbitrated dispute
tree (with $w$-root).

2. If $(\theta, \langle \cdot \rangle, \delta) \notin G$ and $o_{\phi} = \delta$, then there exists an arbitrated dispute
tree (with $L$-root).

**Proof.** In each case, the arbitrated dispute tree $T$ can be
constructed as follows.

1. Let $(\theta, \langle \cdot \rangle, \delta) \in G$ label the $w$-root. For every $a \in Arg_{\phi}$
such that $a \neg \neg \neg \neg \phi (\theta, \langle \cdot \rangle, \delta)$, add a child $L$-node labelled by $a$. Since $(\theta, \langle \cdot \rangle, \delta)$ is in
$G$, for every $a$ with $a \neg \neg \neg \neg \phi (\theta, \langle \cdot \rangle, \delta)$ we find $b \in G$ such that
$b \neg \neg \neg \neg a$. Then, for every $L$-node, we can add a child $w$-node
labelled by some such $b$. The same process can then be repeated
for any $b \in G$ until $G$ is exhausted (guaranteed since $AF(CB, \delta, \phi)$ is
cyclic, cf. Proposition 3.2).

2. Let $(\theta, \langle \cdot \rangle, \delta) \notin G$ label the $L$-root. Since $(\theta, \langle \cdot \rangle, \delta)$ is not in $G$
and $AF(CB, \delta, \phi)$ is cyclic (and thus $G$ is stable), there must be
$a \in G$ such that $a \neg \neg \neg \neg \phi (\theta, \langle \cdot \rangle, \delta)$. We can thus add a child $w$-
node labelled by $a$. Then, for every $c \in Arg_{\phi}$, add a child $L$-
node labelled by $c$, and note that as $a \in G$, for every such $c$
there must be $b \in G$ with $b \neg \neg \neg \neg a$, whence a $w$-child labelled by
$b$ can be added. As in the previous case, this can be repeated
until $G$ is exhausted. ■

In other words, whenever $(\theta, \langle \cdot \rangle, \delta)$ belongs to the
grounded extension, then an arbitrated dispute tree with $w$ success-
fully arguing for $\delta$ can be constructed; and vice versa, whenever $(\theta, \langle \cdot \rangle, \delta)$ does not belong to the grounded extension, then an arbitrated dispute tree with $w$ success-
fully arguing against $\delta$ can be constructed. Again, we believe that these are desirable properties, for
intuitively, if the default is the outcome and it is well-defended
(i.e. grounded), then there should be a way to successfully argue
for it; similarly, if the default is not the outcome and it does not
belong to the grounded extension, then there should be a way to
successfully argue against it. This is in line with existing works on
properties of argumentation, e.g. (Baroni, Rago, & Toni, 2019),
indicating that better argued-for arguments should be stronger than
less well argued-for arguments. Note that Proposition 3.4 gives
sufficient conditions for the existence of arbitrated dispute trees.
However, their existence is not in general guaranteed, as the fol-
lowing example shows.

**Example 3.10.** Consider the case base $CB = \{C_1, C_2, C_3\}$ and the
focus case $\phi$ given below:

$$
C_1 = \langle \{a\}, \langle s_1 \rangle, + \rangle, \quad C_2 = \langle \{a, b\}, \langle s_1, s_2, s_3 \rangle, - \rangle, \quad \phi = \langle \{a, b, c\}, \langle s_1, s_2, s_3 \rangle, - \rangle.
$$

Let $C_0 = (\theta, \langle \cdot \rangle, -)$ be the default case. Now consider
$AF(CB, -, \phi)$. Then, we have $C_0 \rightarrow C_2$, $C_2 \rightarrow C_1$, $C_1 \rightarrow C_0$ and no
other attacks. Thus, the grounded extension of $AF(CB, -, \phi)$ is
$G = \{\phi, C_1, C_2\}$. No arbitrated dispute tree $T$ exists with respect to
$AF(CB, -, \phi)$. Indeed, the root of $T$ would need to be a $w$-node
labelled by $C_0$, which would have an $L$-child labelled by $C_1$, followed by
a $w$-child labelled by $C_2$ and its $L$-child labelled by $C_3$, where
the latter would be a leaf, since $C_3$ is unattacked; but this is not
allowed by (point 4. in) Definition 3.7. Note that the non-existence of an arbitrated dispute tree in this example is consistent with Proposition 3.4, given that $C_0$ $\not\in$ $\mathbb{G}$ but the outcome of the focus case is the default: $o_{\phi} = -$.

In general, situations in which the grounded extension ‘disagrees’ with the arbiter default and focus cases entail that we cannot straightforwardly extract explanations from the corresponding argumentation framework and that possibly some information is lacking in the case base to accordingly explain the outcome of the focus case. We leave the issue of addressing such theoretical situations for future work. In our case study, we encountered just a few such practical situations, specifically due to the lack of (quality) data.

Arbitrated dispute trees give an argumentative reading of cases and relationships among them which explains how the winner defends their position in the dispute. However, in an arbitrated dispute tree we can also identify the excess features possessed by loser’s arguments which the arbiter focus case attacks thus allowing the winner to have the last word in the dispute. Intuitively, those are the features possessed by the parents of the focus case but not the focus case itself, if the focus case appears in the arbitrated dispute tree at all. We formalise this concept next.

**Definition 3.11.** Let $A(F(CB, \delta, \phi))$ and an arbitrated dispute tree $T$ be given. The excess features of $T$ are given by the set

$$\mathcal{F} = \bigcup \{ F \setminus F_0 : [w : (F_0, S_0, o_0)] \text{ is a leaf in } T \text{ with parent } [L : (F, S, o)] \}.$$  

(By convention, if $(F_0, S_0, o_0)$ does not label any node in $T$, then $\mathcal{F} = \emptyset$.)

**Example 3.12.** The arbitrated dispute tree $T_1$ from Example 3.8 has only one leaf $[w : \phi_1 = \{a, c\}, \{s_1, s_2\}, -]$ labelled by $\phi_1$ (see Fig. 4). Its parent is $[L : C_2 = \{b\}, \{s_1, s_2\}, +]$. The difference between the features of $C_2$ and $\phi_1$ is $\{b\} \setminus \{a, c\} = \{b\}$. So the excess features of $T_1$ are given by $\mathcal{F} = \bigcup \{ \{b\} \} = \{b\}$. The arbitrated dispute tree $T_1'$ from Example 3.8 has two leaves labelled by $\phi_1$ (see Fig. 5). Their parents are labelled by $C_4 = \{(a, b), \{s_1, s_2\}, +\}$ and $C_5 = \{(b), \{s_1, s_2\}, +\}$ respectively. The excess features of $T_1'$ are thus given by $\mathcal{F} = \bigcup \{ \{b\} \} = \{b\}$.

The excess features of either $T_1$ or $T_1'$ indicate why the arbiter focus case discards the arguments put forward by $L$. Indeed, in both disputes $w$ is able to counter-argue $L$‘s arguments due to them relying on the feature $b$.

Generally, exposing the excess features of an arbitrated dispute tree, if any, clarifies why some cases do not hold ground in explaining the outcome of the focus case. However, if an arbitrated dispute tree does not have excess features—as, for instance, $T_2$ in Fig. 6 from Example 3.8, then this indicates that all the features possessed by the cases (labelling the nodes) in the tree are to the point.

Note that one could similarly define ‘excess stages’ of an arbitrated dispute tree. However, the distinction between features and stages is that features represent the information that is known about the case at a given stage, whereas stages represent the case’s progress through a (pre-determined) timeline. Therefore, the difference between cases in terms of stages is more of an indication of the possibility in the change of the features (of the ‘lagging’ case) rather than a determinative factor itself. This is especially prominent in our case study whereby a case not progressing through stages is not a factor in itself but merely indicates the lack of information (features) known or recorded about the case. (See e.g. Fig. 10 and discussion later in Section 4.4.)

Having defined and illustrated arbitrated dispute trees as well as excess features, we now formally define explanations of the outcomes of focus cases.

**Definition 3.13.** Let $A(F(CB, \delta, \phi))$ be given. An explanation of the outcome $o_0$ is a pair $(T, \mathcal{F})$ with

- $T$ an arbitrated dispute tree (with respect to $A(F(CB, \delta, \phi))$), and
- $\mathcal{F}$ the excess features of $T$.

**Example 3.14.** Explanations of the outcome $o_{\phi_1} = -$ of the focus case $\phi_1$ from Example 3.2 are $(T_1, \{b\})$ and $(T_1', \{b\})$, where $T_1$ and $T_1'$ are arbitrated dispute trees from Example 3.8, depicted in Figs. 4 and 5, respectively. A unique explanation of the outcome $o_{\phi_2} = +$ of the focus case $\phi_2$ from Example 3.2 is $(T_2, \emptyset)$, where $T_2$ is an arbitrated dispute tree from Example 3.8, depicted in Fig. 6.

The following property concerning existence of explanations follows from the existence of arbitrated dispute trees.

**Corollary 3.5.** Let $G$ be the grounded extension of $A(F(CB, \delta, \phi))$.

1. If $(\phi, \{\}, \delta) \in G$ and $o_0 = \delta$, then there exists an explanation of the outcome $o_0$.

2. If $(\phi, \{\}, \delta) \not\in G$ and $o_0 = \delta$, then there exists an explanation of the outcome $o_0$.

**Proof.** By Proposition 3.4, if the antecedents of either statement are satisfied, then there is an arbitrated dispute tree $T$. Since the excess features of $T$ always exist by definition, clearly an explanation $(T, \mathcal{F})$ exists too.

Having advanced the methodology of our approach to explaining case outcomes, we set forth an application for it in the next section.

4. Case study

We present a case study for the application of our CBR-inspired and AA-driven methodology to the real-world setting of UK primary legislation. Specifically, we focus on the task of explaining the passage (presence or absence of the Royal Assent) of UK Parliamentary Bills in terms of their content-agnostic features and the stages in the Parliament’s rules of conduct. We show how Parliamentary Bills instantiate cases and how their outcomes can be explained through arbitrated argumentative disputes. The purpose of explaining Bill outcomes is to inform understanding of the factors reflecting and possibly influencing the passage of Bills. We develop a system integrating our methodology and visualisations thereof with the purpose of evaluating and showcasing our methodology for yielding automated explanations. The system provides a collaborative environment to enhance interactive discussions on a large scale.

We have implemented a system that collects publicly available data about Parliamentary Bills, structures it for internal representation via cases and argumentation frameworks (AFs), and visualises arbitrated dispute trees thus yielding explanations for Bill outcomes. We briefly review data collection in Section 4.1. We then show how Bills instantiate cases in Section 4.2. We explore arbitrated argumentative disputes and explanations using visualisations in Section 4.4. Finally, we discuss the findings and evaluate the application against legislative expert opinion in Section 4.5.

4.1. Data collection

The main data source used was the Parliament’s database.¹ We utilised the following additional sources of information: the

¹ http://explore.data.parliament.uk/.
Parliament’s website;\(^2\) the charitable website TheyWorkForYou;\(^3\) the not-for-profit website The Public Whip.\(^4\) We collected data covering Bills dating from 2005-05-11 to 2017-05-03, corresponding to sessions from 2005–2006 to 2016–2017, as available on the Parliament’s database.\(^5\) This yielded 1754 Bills.

Unfortunately, the Parliament’s database and website did not have high quality data: not only it was often incomplete and internally inconsistent, but also lacked in granularity and detail. Nonetheless, we used the other information sources to complement the information available on the Parliament’s database and to identify additional features of Bills. In the end, we fixed major issues together with the legal experts and successfully prepared the case study.

The collected features can be classified as follows, for each Bill:

- Nominal information about the Bill: a) type, b) starting House of Parliament, c) number of sponsors, d) ballot number (where applicable), e) type of Committee (where applicable), f) absence or presence of the Royal Assent (RA);
- Stages of the Bill, i.e. dates of Bill’s readings.

Extensive information on Bills and their passage can be found on the Parliament’s web pages.\(^6\) Here, we provide a brief summary of the main aspects, relevant to our case study. The type of a Bill reflects the importance and the origin (among other things, such as coverage of the population and ways of introduction) of the Bill. For instance, Government Bills are introduced by the presiding Government and are arguably the most important in terms of considerations and changing the law as it applies to the whole society; Private Bills are usually promoted by organisations, such as local authorities or private companies, and often are of importance only to specific individuals or organisations. Bill types also determine the rules of conduct: different rules regarding time and priority for debating the Bills apply depending of Bills’ type. For instance, Bill Ballots are chosen in a ballot once per Parliament session and are given a priority ranking (number) which somewhat determines how much time there will be to debate them: as a rule of thumb, the lower the number, the better; Top 7 Bills are often considered to be the most likely to pass.

Bills can start in either the House of Commons (also Commons, or HoC) or the House of Lords (also Lords, or HoL), but need to go through debates in both in order to be passed as law (i.e. be granted Royal Assent). Largely financial Bills start in Commons, whereas HoL Private can be introduced only in Lords. The two Houses also have complicated codes of conduct of business, and this influences the passage of Bills. Number of Bill sponsors records nominally the person(s) sponsoring the Bill when introduced. Different committees scrutinising various aspects of a Bill are gathered differently in the two Houses. Different type of committees are generally harder (Opposed) or easier (Unopposed) to pass through.

The debates happen in stages, called readings, that take place in both Houses, with at most three readings happening first in the starting House, and then possibly continuing through at most three readings in the other House, whence finally the debated Bill may be granted Royal Assent.

4.2. Instantiating cases with bills

To analyse Bills using our methodology, we first instantiate cases with Bills as follows. We represent every Bill as a case consisting of a set of features, a sequence of stages and an outcome. We use the following features based on Bill nominal information.

- a) Type: Government, Hybrid, Private, Ballot, Ten Minute Rule, Presentation, HoL Private;\(^7\)
- b) Starting House of Parliament: House of Commons (HoC), House of Lords (HoL);
- c) Number of Bill’s sponsors: 1, 2;
- d) Ballot number (where applicable): natural numbers from 1 to 20, binarised into whether it is Top 7 (i.e. 1,…,7) or not;
- e) Type of Committee (where applicable): Opposed, Unopposed.

So \( F = \{\text{Government, Hybrid, Private, Ballot, Ten Minute Rule, Presentation, HoL Private}\} \) is the set of features.

Note that any Bill is of one and only one type. Hence, any case will have one and only one feature from \{Government, Hybrid, Private, Ballot, Ten Minute Rule, Presentation, HoL Private\}. We can thus partition the Bill dataset by Bill type (for readability as well as implementation efficiency purposes). That is, we will consider different case bases for Bills of different types, where all the Bills instantiating cases in a case base will be of the same type. Note also that as any Bill is of exactly one type, when partitioning Bills by type we can still assume the whole \( F \) as the feature set, because for a given Bill type, none of the other types will appear as features in cases.

In terms of stages of Bills, we use the following ones.

1. 1\(^{st}\) House 1\(^{st}\) Reading (H1R1),
2. 1\(^{st}\) House 2\(^{nd}\) Reading (H1R2),
3. 1\(^{st}\) House 3\(^{rd}\) Reading (H1R3),
4. 2\(^{nd}\) House 1\(^{st}\) Reading (H2R1),
5. 2\(^{nd}\) House 2\(^{nd}\) Reading (H2R2),
6. 2\(^{nd}\) House 3\(^{rd}\) Reading (H2R3),
7. Royal Assent (RA).

So \( S = \{H1R1, H1R2, H1R3, H2R1, H2R2, H2R3, RA\} \) is the sequence of stages.

Regarding the outcomes, the Bills can either be Accepted or Rejected. A Bill is Accepted only when it reaches the last stage RA (i.e. it is granted the Royal Assent and becomes law). Otherwise, any Bill that does not reach the RA stage is Rejected (i.e. it does not become law). In particular, the outcome of any Bill at any stage is Rejected unless and until it reaches the RA stage whence it outcome flips to Accepted. So \( \emptyset = \{\text{Accepted, Rejected}\} \) is the set of outcomes.

Concerning the defaults, we note that from the legislative point of view, Bills of different types adhere to different rules of conduct in the Parliament. For instance, Government Bills are always given time to be discussed at length, whereas Presentation Bills are often merely introduced to the Parliament but rarely discussed in depth. As a consequence, Bills of different types carry different inherent biases towards their outcomes. For instance, Government Bills are very likely to pass, whereas Presentation Bills rarely pass. To reflect this we set different default outcomes for different type of Bills as follows.

- Government: Accepted;
- Hybrid: Accepted;
- Private: Accepted;
- Ballot: Accepted;
Ten Minute Rule: Rejected;
Presentation: Rejected;
Hol. Private: Rejected.

Overall, our setting is as follows:
• \( F = \{ \text{Government, Hybrid, Private, Ballot, Ten Minute Rule, Presentation, Hol. Private, HoC, Hol, Sponsors1, Sponsors2, Top 7, Opposed, Unopposed} \} \);
• \( S = \{ \text{H1R1, H1R2, H1R3, H2R1, H2R2, H2R3, RA} \} \);
• \( \Omega = \{ \text{Accepted, Rejected} \} \) with
  - default Accepted for focus cases \((F_\phi, S_\phi, o_\phi)\) such that \(F_\phi \cap \{ \text{Government, Hybrid, Private, Ballot} \} \neq \emptyset\),
  - default Rejected for focus cases \((F_\phi, S_\phi, o_\phi)\) such that \(F_\phi \cap \{ \text{Ten Minute Rule, Presentation, Hol. Private} \} \neq \emptyset\).

Note again that as each Bill is of exactly one type, the default outcomes are well-defined.

In effect, we construct 7 case bases \(CB_t\) for \(t \in \{G, H, Prv, Blt, TMR, Pre, HoL\}\) with default outcome Accepted for \(t \in \{G, H, Prv, Blt\}\) and default outcome Rejected for \(t \in \{TMR, Pre, HoL\}\). We call these Parliamentary case bases. The sizes of the case bases are in the hundreds, except for \(|CB_H| = 3\) and \(|CB_{HoL}| = 31\).

To establish that Parliamentary case bases are well-defined according to Definition 3.1, we show that they are coherent.

Proposition 4.1. A Parliamentary case base \(CB_t\) is coherent for every \(t \in \{G, H, Prv, Blt, TMR, Pre, HoL\}\).

Proof. Fix \(t \in \{G, H, Prv, Blt, TMR, Pre, HoL\}\), let \((F, S, o), (F', S', o') \in CB_t\) and suppose \(F = F'\). There are two possibilities for \(S = S'\).

1. If \(S = S' = (s_1, \ldots, s_m)\) for some \(m < n\) (where \(s_1 = \text{H1R1}\) and \(s_n = \text{RA}\)), then \(o = \text{Rejected}\) and \(o' = \text{Rejected}\), whence \(o = o'\).

2. If \(S = S' = (s_1, \ldots, s_n)\), then \(o = \text{Accepted}\) and \(o' = \text{Accepted}\), whence \(o = o'\).

In any event, \(CB_t\) is coherent. \(\square\)

Having instantiated cases with Parliamentary Bills, we can use our methodology from Section 3 to map Parliamentary case bases to AFs and to explain the outcomes of selected focus cases. Before that, though, we discuss the visualisation methodology that we used to visualise arbitrated argumentative disputes so as to facilitate the explanation process.

4.3. Visualising arbitrated argumentative disputes

Visualisation is the process of forming a mental model of data in the mind of the beholder (Spence, 2014). In our context we visualise data to help the reader understand our explanation. To this end it is important to present the viewer with both the high level picture of the argumentation and with details which will help in its comprehension. At a high level we show all of the Bills appearing in the arbitrated dispute tree, see Fig. 7. In more detail, each Bill is titled and presented as a list of features and stages on the left and right, respectively, within each node. Red and green background under the title indicate the Rejected and Accepted outcomes, respectively. The Bill whose outcome is being explained is highlighted with a pink rounded frame, whereas the default case (titled ‘Default’) is highlighted with a yellow rounded frame. The attacks are represented as arrows, highlighted in orange.

We developed (precursors of) the visualisations for large scale collaborative viewing on a Large High Resolution Display (LHRD) at Imperial College London Data Science Institute, see Fig. 8. This provided a 2.4 × 16 m circular discussion environment for experts and non-experts alike to view, understand and form hypotheses about the Bill outcomes and their explanations. The scale of the environment provided a convenient mechanism to appreciate large arbitrated dispute trees with tens (and even hundreds) of nodes while still providing details about features and stages. The social nature of such environments for the collaborative exploration of data provoked discussion on the implications of the argumentative explanations and the assumptions of the viewers which is hard to replicate in a traditional presentation. See Fig. 9 for a snapshot of a live discussion concerning UK Parliament Bills.

To render visualisations, we created graph-based visualisation tools using C#, Windows Presentation Foundation and the GraphSharp Library. As the order of the nodes in arbitrated dispute trees is hierarchical and tree-like, we used an efficient version of Sugiyama’s Algorithm (Eiglsperger, Siebenhaller, & Kaufmann, 2005) to control the graph layout. This layout algorithm is scalable to many hundreds of nodes should the number of cases be as large.

---


4.4. Explaining bill outcomes via arbitrated argumentative disputes

To illustrate explanations of Bill outcomes, we consider several cases and give visualisations of arbitrated dispute trees and identify the excess features of the arbitrated dispute trees. We then discuss the insights stemming from our explanations.

Fig. 7 shows an arbitrated dispute tree $T_{pre}$, with respect to the argumentation framework $AF(C_{pre}, Rejected, \phi_{pre})$, where the focus case $\phi_{pre} = \{(Presentation, HoC, Sponsors1), (H1R1, Rejected)\}$ is the Wild Animals in Circuses (Prohibition) Bill. The focus case $\phi_{pre}$ and the default case ($\phi, \emptyset$, Rejected) act as arbiters and assign the roles of the winner $\emptyset$ and the loser $\phi$: $\emptyset$ is arguing for the default outcome Rejected, and $\phi$ for Accepted. The only $\emptyset$’s argument pertains to the Leasehold Reform (Amendment) Bill, which is irrelevant due to the feature Sponsors2, and is arbitrated away by $\phi_{pre}$. $\emptyset$ thus wins the dispute and the outcome Rejected holds ground.

The excess features of $T_{pre}$ are $F_{pre} = \{Sponsors2\}$. An explanation of the outcome Rejected for this Bill is thus $T_{pre}, F_{pre}$. The fact that the Wild Animals in Circuses (Prohibition) Bill had only one sponsor rather than two indicates that the Bill may possess unrecorded factors that are substantively determinative of the Bill’s outcome, as discussed in Section 4.5.

Fig. 10 shows an arbitrated dispute tree $T_{hol}$, with respect to the argumentation framework $AF(C_{hol}, Rejected, \phi_{hol})$, where the focus case $\phi_{hol} = \{(HoL Private, HoL, Sponsors2), (H1R1, Rejected)\}$ is the Modern Slavery (Transparency in Supply Chains) Bill. The excess features of $T_{hol}$ are $\emptyset$. The explanation $T_{hol}, F_{hol}$ indicates that the Bill in question failed to pass to stages further than H2R1, which may be indicative of unrecorded factors substantively determinative of the Bill’s outcome, again as discussed in Section 4.5.

Fig. 11 shows an arbitrated dispute tree $T_{G}$, with respect to the argumentation framework $AF(C_{G}, Accepted, \phi_{G})$ where the focus case $\phi_{G} = \{Government, HoC, Sponsors2, H1R1, H1R2, H1R3, H2R1, H2R2, H2R3, RA\}$ is the Policing and Crime Bill. The excess features of $T_{G}$ are $F_{G} = \{Sponsors1\}$. The explanation of the outcome Accepted is $T_{G}, F_{G}$.

While the above examples illustrate arbitrated dispute trees of small size (several nodes), Fig. 12 shows that our system accommodates larger arbitrated dispute trees too (cf. Section 4.3).

4.5. Expert evaluation

The adequacy and outputs of the approach to the application were considered by the group of experts in legislation who had been involved in determining the parameters of the data input, thoroughly discussing the methodology as well as evaluating the presentation of the approach: Rajvinder Dulay and Sally Turvey, Thomson Reuters (TR) experts in legislative processing and publishing; Daniel Greenberg, General Editor of Thomson Reuters Westlaw UK Annotated Statutes and Counsel for Domestic Legislation, House of Commons; Tharindu Haruarachchi who provided data science expertise from TR Labs (London).

Overall, the legislative experts have evaluated the system and its outputs highly positively as providing novel and significant insights into the analysis of UK primary legislation. On the one hand, the experts were able to confirm that for the most part Bill analysis and output explanations matched what would be expected as a matter of political and legislative theory and practice. This is a positive evaluation as it confirms empirically that our methodology and its application to batch-processing large numbers of Bills are sound. On the other hand, explanations of Bill outcomes via arbitrated argumentative disputes helped the legislative experts to assist in their decision-making.
understand the proposed methodology and what it amounts to in the context of Parliamentary Bills. The visualisation was a particularly useful step to help explain the methodology to lay people showing real-life examples of Bills and their relationships in a graph, and to walk through the graphs reading them as disputes. In addition, the application of our methodology to Parliamentary Bills and the resulting explanations with the excess features of arbitrated dispute trees have helped to identify some unexpected factors concerning Bill outcomes. We discuss these next.

First, explanations of Bill outcomes identified that the number of Members of Parliament sponsoring a Bill is significant. In one sense this could be dismissed as a ‘false result’, as the number of sponsors is determined in ways that make it impossible for it to be a determinative feature in practice of whether or not a Bill will succeed in achieving Royal Assent. Simply dismissing this finding, however, would be failing to pay attention to its possible implications: the output shows sponsor numbers as a clear indicator, and the next research phase should therefore be to consider whether sponsor numbers as interpreted and recorded by the data input are indicative of some other feature that is substantively determinative, such as the influence of cross-party support in relation to a Private Member’s Bill.\[11\] This unexpected result is therefore one

---

11 Ballot, Ten Minute Rule, Presentation and HoL Private Bills are collectively called Private Member’s Bills; in contrast, Government Bills are called Public Bills.
of the most exciting aspects of the application outputs from the perspective of the legislative expert.

Similarly, explanations of Bill outcomes recorded an interesting, and substantively counter-intuitive, finding in relation to the ballot numbers of (Private Members') Ballot Bills in the House of Commons. Twenty successful members are drawn from the ballot in each Parliamentary Session, but the way in which time on the Order Paper on Private Members' Fridays is allocated by Standing Orders means that the Top 7 in the ballot (those numbered 1-7) have a significantly greater chance of achieving Royal Assent than the remaining thirteen. The application, however, showed that at least within the historical batch of data analysed, being within the Top 7 is not crucial to explaining the outcome of a Bill. It is possible to speculate on substantive political reasons for this finding, including the well-known practice of the Government of handing out certain Bills to Private Members and providing support for those Bills, which therefore become in effect Government Bills masquerading as Ballot Bills. But there may be other presently hidden factors that distort the obvious implications of the Standing Orders. So again, merely dismissing this as a ‘false’ result would be to fail to appreciate the potential analytical implications of the methodology when taken further in terms of the application.

Overall, therefore, the legislative experts confirmed the outputs provided by our methodology as being extremely interesting in confirming known patterns for the most part, and as being of potential significance if taken further into the analysis of those patterns which deviate from what would be expected as a matter of substantive consideration by reference to already known criteria. The methodological explanations of Bill outcomes—encompassing arbitrated dispute trees, as well as their excess features, indicating sometimes unexpected features—were crucial in determining those patterns. The means of presentation by showing real-life examples of Bills graphically (i.e. visualising an argument graph) with common or opposing features were pivotal in detailing and evaluating the methodology of our approach. Subject matter experts appreciated the contrastive method of presenting Bills with diverging outcomes in explanations, the possibility to select how many explanations and/or Bills to display, as well as the social nature of how the explanations played out. (See also Section 5.5 for a discussion on the latter three points.)

5. Related work

This paper touches on several strands of related work in AI: explainable argumentative reasoning; explanations as argument graphs; explanations mined from, and relating to data; evaluating explanations. We discuss these in what follows.

5.1. Explainable reasoning using argumentation

Our work presents a development in the line or works that instantiate abstract argumentation (AA) with cases for the purposes of reasoning and explanation. In particular, Çyras, Satoh, and Toni (2016a,b) proposed the formalism called AA-CBR that integrates case-based reasoning (CBR) representation with the reasoning mechanisms of AA for the task of deciding outcomes of new cases based on past cases. The authors also suggested how the outcomes obtained using AA-CBR could be explained via (variants of) the dispute trees defined in (Dung, Kowalski, & Toni, 2006; Dung, Mancarella, & Toni, 2007) extracted from the argument graphs corresponding to case bases. Still further, Cocarascu et al. (2018) integrated AA-CBR with a neural network architecture so that the latter helps to reduce the number of features to be considered.
when predicting outcomes, given high-dimensional data representation. The authors also proposed to generate, from the argument graphs, logic programs which yield outcomes of new cases that provably equivalent to AA-CBR outcomes and also act as rule-based explanations of thus established outcomes.

From the point of view of knowledge representation, our approach is more general than AA-CBR. Specifically, in AA-CBR cases consist of a set of features and an outcome. In contrast, we consider cases comprising the additional component of stages so as to model a more complex setting where the outcomes can change from stage to stage even when the features remain the same. In this respect, our approach conservatively extends AA-CBR in the sense that if the stages component is omitted (and in particular Definitions 3.3 and 3.5 use only the features and outcomes of cases), then we obtain the definitions of AA-CBR argument graphs. However, in spite of generalising AA-CBR, our approach exhibits several crucial differences with respect to the works mentioned above.

In particular, we are not concerned with deciding or predicting outcomes of new cases, but rather explaining outcomes of cases already decided by either humans or machines. To this end, we understand explanations through arbitrated argumentative disputes, where the arbitrator focus and default cases play the pivotal roles of structuring the dispute in a way that the winner of the dispute argues for the outcome to be explained. This also ensures that winner always has the last word in the dispute (leaves of arbitrated dispute trees are w-nodes) and, unlike in AA-CBR explanations, the defence set (arguments labelling w-nodes) is always contained in the grounded extension of the underlying argument graph.

More generally, although arbitrated dispute trees share some aspects of the dispute trees used as explanations in AA-CBR, they differ in some important aspects.  

1. in dispute trees the root can be labelled by any argument, whereas in arbitrated dispute trees the root is labelled by the default argument;
2. in dispute trees the root is necessarily a w-node, whereas in arbitrated dispute trees it is either a w- or an L-node;
3. in dispute trees every L-node has at most one child, as opposed to exactly one child in arbitrated dispute trees.

Other additional conditions are often imposed on dispute trees to capture desirable properties, for example that no argument labels both w and L nodes. We do not need to impose additional conditions because argument graphs in the context of cases are acyclic. Instead, taking the focus case and the default case to act as arbiters and decide whether the root is a w or an L node allows arbitrated dispute trees, and consequently explanations, to exhibit various desirable properties without additional constraints on the structure of the trees. We leave it for future work to study whether similar concepts can be defined for dispute trees in AA-CBR and what properties they exhibit.

5.2. Explanations as argument subgraphs

A principal component of our explanations is the concept of an arbitrated dispute tree, effectively a subgraph of an AF seen as an argument graph. Several works define methods for determining explanations for the (non-)acceptability of arguments in argument graphs, e.g. (Fan & Toni, 2015a; 2015b; Schulz & Toni, 2013; Zhong, Fan, Luo, & Toni, 2019). These works essentially use dispute trees as the underlying mechanism for computing explanations, and so differ from our approach due to reasons discussed in regards to AA-CBR (Section 5.1).

Instead, different type-of subgraphs of an argument graph are considered as the basis of explanations for the acceptance of arguments by García et al. (2013).

They formalise dialectical explanations of argument acceptance (or ‘warrant’) in the context of AA, by defining explanations as triples of sets of argument graphs, called dialectical trees. The respective elements of the explanation triple consist of graphs standing intuitively for, against and neither with respect to the acceptance of a designated claim. The authors define claims of arguments abstractly, and assume existence of a function (subsuming classical negation) that identifies contradictory claims. This is akin to the outcomes δ and δ comprising Ω in our approach.

There are several key differences between the work of García et al. (2013) and our work. First, a dialectical tree is a subgraph on the argument graph such that each of its paths satisfies the given dialectical constraints. For instance, the non-circularity constraint essentially prevents directed cycles in the argument graphs. In contrast, we obtain argument graphs from case bases that do not need to fulfill any further constraints, e.g. we prove that they are acyclic in general.

Further, there can generally exist dialectical trees for both a claim and its contradictory claim. In contrast, arbitrated dispute trees with respect to a given argument graph are rooted in either w- or L-nodes exclusively, depending on the outcome of the focus case. Also, dialectical trees are often subtrees of another; thus possibly having leaves attacked by other arguments. In contrast, arbitrated dispute trees always have leaves labelled by unattacked arguments, representing the end of the dispute.

Importantly, García et al. (2013) focus on AA where arguments are not instantiated, treating arguments as atomic entities and taking attacks among them as given. Thus, there is no structure within the arguments. To contrast, in our approach arguments are instantiated with cases. This imparts a certain structure on arguments, which allows to constructively define attacks and structure argument graphs accordingly. In particular, instantiating arguments with cases allows to automatically populate AFs with mined data, thus paving way to relate explanations to the data they are explaining.

5.3. Explanations relating to data

Our approach enables automated instantiation of arguments with, and extraction of explanations from data mined in real-world applications. While logical, e.g. (Cocarascu et al., 2018), and linguistic, e.g. (Zhong et al., 2019), argumentative explanations extracted from mined data exist, to the best of our knowledge argumentative explanations linking back to data do not exist. Our approach provides first-of-a-kind such explanations by incorporating the excess features of arbitrated dispute trees. These indicate why the arbitrator focus case discards the arguments put forward by the loser of the dispute, thus linking back to data features that influence the explained outcome.

Whereas we complement our explanations with the excess features, García et al. (2013) propose a simple query answering methodology relying on their explanations. It takes as input a query a claim and returns a yes, no, undecided or unknown answer depending on whether the claim or its contrary are warranted, or neither. Query answering is not directly relevant to our approach because the outcomes of focus cases are given rather than determined by means of argumentation, but it would nevertheless be interesting to investigate in the future if we could integrate query answering with respect to e.g. features or stages.
5.4. Evaluating explanations

One aspect that allows our work to stand out among those discussed in this section is our application to a real-world problem and the resulting expert evaluation of the practicality of our methodology. There are some other works in argumentation, e.g. (Cerutti, Tintarev, & Oren, 2014; Rosenfeld & Kraus, 2016), that investigate the usefulness of explanations in argumentation with users, but these typically do not relate to mined real-world data. Zhong et al. (2019) is a recent work where argumentation is used to explain multi-criteria decision making via natural language explanations generated from dispute trees, and the usefulness of explanations is evaluated in an empirical study with experts analysing legal cases. We have instead evaluated our approach in a legislative setting by means of legal expert opinion as well as collaborative exploration of Bill outcome explanations. We will study in the future if and how our explanations can be rendered in natural language and aim to conduct a structured empirical study.

5.5. Explanations in Al

Explainability is a hot topic in AI at present, e.g. for machine learning (Biran & Cotton, 2017; Costabello et al., 2019) and other forms of AI (e.g. scheduling Cyras, Letsios, Misener, & Toni, 2019). In a recent survey on relating philosophical, social and cognitive science works to explainable AI, Miller (2019) highlights the following major findings relevant to our study: explanations are contrastive, selected and social. Being contrastive means that explanations address the question ‘why that particular outcome?’ by contrasting with ‘why not some other outcome?’ Being selected means that not necessarily all the explanatory reasons are sought after, but only some of them, possibly only one. Being social means that explanations transfer knowledge, often in an interactive and conversational manner. Our explanations via arbitrated argumentative disputes can be seen to possess these properties.

First, arbitrated dispute trees contrast cases with diverging outcomes. They also relate diverging cases in terms of their features and stages, by their specificity and concision, as well as advance and proximity, respectively. The explanations thus show what features and stages cases have in contrast to the focus case. Excess features in particular reveal contrast between the focus case and cases ‘similar’ to it. Our explanations are thus contrastive in considering the question ‘why this but not the other outcome?’

Second, our explanations are selected, because in general there can be multiple arbitrated dispute trees (and the excess features) that explain the outcome of the focus case. In other words, when there are multiple ways for the winner to win the dispute, each one of those suffices as an explanation. While in practice the selection process is influenced by cognitive biases (Miller, 2019), in theory one would like to maintain all explanations and let the user express preferences pertaining to the selection of explanations. We leave modelling of such preferences and election mechanisms for future work.

Third, methodologically and presentation-wise our explanations are social. On the one hand, arbitrated dispute trees model a conversational dispute.13 On the other hand, in the case study, we presented our explanations visually and interactively in a collaborative environment. Both these aspects serve the purpose of knowledge transfer in explaining the outcomes of focus cases.

Overall, by exhibiting the above properties, explanations via arbitrated argumentative dispute are contextualised (Miller, 2019) in the setting of an application, allowing for a rich discussion about a particular case study. Especially concerning UK primary legislation, our explanations drew attention to unexpected patterns of Bill passage, indicating non-causal factors of interest, and afforded well-appreciated debates and insights into the topic. We leave it for future work to further study the more philosophical aspects of the explanations and the underlying argumentative methodology presented in this paper.

Argumentation is but one area of research suitable for explaining AI (Moulin, Irandoust, Bélanger, & Desbordes, 2002): from explaining decision making, e.g. (Amgoud & Prade, 2009; Zeng et al., 2018; Zhong et al., 2019), to explaining Bayesian networks (Timmer, Meyer, Prakken, Renooij, & Verheij, 2017), scheduling (Čyras et al., 2019), classifications (Amgoud & Serrurier, 2008), predictions (Cocarascu et al., 2018), recommendations (Rago et al., 2018). Recent research in explaining AI also concerns symbolic approaches for explaining Bayesian networks (Shih, Choi, & Darwiche, 2018), counterfactual natural language statements for explaining predictions (Sokol & Flach, 2018), explainable planning (Fox, Long, & Magazzeni, 2017), and generally explainable machine learning (see e.g. (Biran & Cotton, 2017) for an overview) as well as integration of model-based and model-free approaches for explainable AI (Geffner, 2018). It would be interesting to study how our approach could be integrated with model-free learning approaches for data mining and applied to e.g. classification and prediction tasks.

Argumentation has been widely applied in legal reasoning, often with the purposes of aiding and automating reasoning (see Bongiovanni et al., 2018) for a recent overview of legal reasoning and argumentation). Some works in that area use argumentation for explainable legal reasoning. For instance, Al-Abdulkarim, Atkinson, and Bench-Capon (2016) use abstract dialectical frameworks (a form of argumentation) to aid with design, implementation and explainability of automated legal case-based reasoning tools. Vlek, Prakken, Renooij, and Verheij (2016) and Timmer et al. (2017) use argumentation to facilitate explanation of the inner workings of a Bayesian network used for probabilistic legal reasoning. Gordon and Walton (2012) use argumentation to support argument construction and evaluation, and consequently decision explanation in legal cases. Our case study in particular falls in the broad area of legal reasoning and argumentation. However, its focus is on explaining legislation outcomes rather than automating reasoning with case law.

6. Summary and future work

In this paper we put forward a methodology (inspired, loosely speaking, by case-based reasoning (CBR) and driven by abstract argumentation (AA)) for explaining outcomes of cases. Specifically, we defined a general setting of (given and focus) cases determined by features, stages and outcomes. We advanced an instantiation of AA frameworks mined from cases, with cases as arguments and dialectical relationships among them determined by diverging outcomes, specificity and concision in terms of features, as well as advance and proximity in terms of stages. We proposed a novel way to explain outcomes of focus cases via arbitrated argumentative disputes using the mined AA frameworks. To that end, we defined arbitrated dispute trees representing disputes between a winner and a loser, arbitrated by the focus (as well as the special default) case. We also defined excess features of dispute trees that explain why the arbiter focus case discards the loser’s arguments in the dispute. We established various desirable properties of our explanations, which encompass arbitrated dispute trees and excess features, and thus go beyond the state-of-the-art in argumentation-based explanation. Finally, we presented a case study for the application of our methodology to the UK primary legislation setting, whereby we instantiated arguments/cases with Parliamentary Bills and explained their outcomes using content-agnostic features.

13 Which need not be in natural language, as pointed out by Miller (2019).
There are many promising future work directions regarding both our methodology and case study, in addition to those mentioned in Sections 4.5 and 5. In terms of the methodology, we plan to study whether non-discrete features could be dealt with too, and also if more than two outcomes could be accounted for. Preliminary investigations indicate positive answers in terms of argumentative representation, but the notion of an arbitrated dispute tree in such a setting needs to be scrutinised. More generally, we will explore further the constituents of explanations and their properties, for instance regarding their minimaliness in terms of arguments, features etc. Explanations for different problems, such as why a given case has not advanced further than it currently has, are also of interest.

In terms of the case study, it would be most interesting to capture more features of different kinds. Specifically from a practical legislation perspective, it would be very interesting to expand the data parameters so as to reflect and explain, and therefore ultimately predict, factors that outweigh the procedural advantage of a Top 7 Ballot Bill, both in terms of disadvantaging those in the Top 7 and in terms of advantaging those in the bottom 13. While this may merely confirm what legislative experts already know in some respects, there is potential to reveal factors and patterns that are presently hidden. More generally, whereas we have experimented with various other nominal features of the bills, such as additional stages and sittings information, it seems that the procedural and contextual features—such as voting information, Parliament’s schedule information, cross-party support, external events—could be a promising source for elaborating explanations.

**Credit authorship contribution statement**

**Kristijonas Ćyras:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing - original draft, Writing - review & editing, Visualization. **David Birch:** Software, Writing - original draft, Visualization. **Francesca Toni:** Conceptualization, Methodology, Software, Formal analysis, Resources, Writing - original draft, Writing - review & editing, Supervision, Project administration, Funding acquisition. **Rajinder Dulay:** Conceptualization, Validation, Resources, Data curation. **Sally Turvey:** Validation, Resources. **Daniel Greenberg:** Conceptualization, Validation, Resources, Data curation, Writing - original draft. **Tharindu Hapuarachchi:** Conceptualization, Validation, Resources, Data curation, Project administration.

**Acknowledgements**

Kristijonas Ćyras, Francesca Toni, David Birch and Yike Guo were partially funded by Thomson Reuters during the period of writing this paper. The authors thank Elna Michaeilidou for helping with the data collection, and Jingwen Bai for helping with the data visualisation. The authors also thank the anonymous reviewers for their valuable feedback and input.

**References**


156


