A Computational Approach to Identifying Formal Fallacy

Gibson A., Rowe G.W, Reed C.
University Of Dundee
aygibson@computing,dundee.ac.uk
growe@computing.dundee.ac.uk
creed@computing.dundee.ac.uk

Abstract

In this article we attempt to lay the groundwork for a computational method of identifying fallacy. We introduce the concept of a fairly simple approach to dealing with identifying traditional formal fallacies computationally and briefly discuss possible future extensions that could be made to such a system once implemented. An example is provided of how one may approach the identification of the fallacy of Affirming the Consequent in a computational setting.

1 Introduction

The main problem we face when we study fallacy is that it is very difficult to define. Aristotle said "That some reasonings are genuine, while others seem to be so but are not" **[Aristotle, 1955]**. [**Hamblin, 1970]** discusses how the definition of a fallacy being an argument which looks valid but is not is the traditional definition of fallacy. van Eemeren and Grootendorst defined fallacy as a move that violates the rules of critical discussion [**van Eemeren and Grootendorst 1984**]. Without a definitive answer to the question "What is a fallacy?" making any use of fallacy in argumentation computationally is difficult and ultimately based on the definition of fallacy we choose.

A classification of fallacy that is widely accepted is the classification of fallacies into two groups "*formal fallacies* and *informal fallacies*". Formal fallacies are arguments that have an invalid logical form with respect to the rules of propositional logic. Informal fallacies are linguistic in nature and are embedded in the context, meaning, and textual content of the argument. The formal fallacies are far simpler to consider from a computational approach than the informal fallacies as they are structural in nature and not heavily reliant on an understanding of the language used in the argument.

Despite an increasing interest accumulating in the study of fallacy, fallacy is a field of argumentation that is still considered under researched. Advances in the understanding and approaches to the use of fallacy and the role it plays in argumentation have been made by [**Hamblin, 1970]**, [**Woods and Walton, 1972], [Mackenzie, 1979]**, [**Walton, 1984]**, [**Walton, 1995]**, [**Godden and Walton, 2005]**, [**Wells and Reed, 2006]**. This research has concerned itself with the theoretical aspects and discussions of the theoretical questions that arise in the study of fallacy. In particular **[Hamblin, 1970]** in response to the traditional treatment of fallacy developed a game ***H*** within a dialectical setting. The dialogue game ***H*** defines a set of rules that dictate the moves that the participants in the dialogue can make. Hamblin also introduced the concept of commitment within the dialogue in that as players within the game ***H*** make their locutions the locutions perform operations on the commitment stores of the players. Hamblin defined a set of rules in the game ***H*** that he believed prohibited the fallacy of Begging the Question from occurring. This was achieved through the rules having dependencies on previous moves made by each player and the contents of the commitment stores of the players on a given turn. This treatment of fallacy through the use of a dialectical game was an original and innovative approach.

[Mackenzie, 1979] extended the dialogue game developed in Fallacies [**Hamblin, 1970]** by introducing two new dialogue games called ***DC*** and ***DD*** respectively. Mackenzie's work was in response to [**Woods and Walton, 1978]** who showed how it was possible for an argument to be made in a dialogue game that conformed to the rules of ***H*** but was still circular in nature and possibly an instance of Begging the Question. Walton & Woods achieved this by exploiting the non cumulative nature of dialogue game ***H***. Non cumulative means that players have the ability to retract commitment to propositions during the course of the dialogue. Walton & Woods argued that the property of cumulativeness was important in a dialogue for the elimination of the fallacy of Begging the Question. The extensions Mackenzie defined in the games ***DC*** and ***DD*** were designed to show how the fallacy of Begging the Question could be eliminated from even non cumulative dialogues through the use of further innovative rules.

Making use of dialogue rules to prohibit the use of fallacy raises the question "Can we construct a dialogue game such that the rules of the game prevent any fallacy occurring?" If we believe the words of Joseph in *Introduction to Logic* who makes the claim "Truth may have its norms, but error is infinite in its aberrations and they cannot be digested in any classification." [**Hamblin, 1970]**, then constructing a game that is able to exclude all instances of fallacy through its complex set of rules could prove as elusive as defining fallacy itself.

[Walton, 1995] introduced a modern approach to the treatment of fallacy. Walton defined fallacy as an argumentation technique that can be used successfully by participants in dialectical engagement. Fallacy is then only considered an invalid argumentation technique when it is used to block the goals of the dialogue the participant is involved in. If fallacy can be considered a valid argumentation technique then rather than construct games in such a manner that we prevent fallacy occurring we should be looking to a method in which we embrace fallacy as part of our dialogue games.

2 Approach

The fallacy of Affirming the Consequent is one of the 13 fallacies identified by Aristotle in his "De sophisticis elenchis" **[Aristotle, 1955]**. We will attempt to define a simple method to allow us to identify the fallacy of Affirming the Consequent in a textual argument.

Affirming the consequent takes the following form [1]:

*1. If **a**, then **b**.*
*2. **b***
*3. Therefore, **a***

1. $a \supset b$
2. b
3. a

An example illustrating an instance of the fallacy of Affirming the Consequent is shown in Figure 1.

"If Bill Gates wins the lottery then he will be a millionaire. Bill Gates is a millionaire therefore Bill Gates won the lottery."

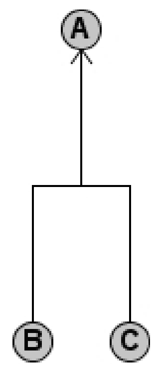
Figure 1.

We can see an explanation of the propositions in Figure 2.

Bill Gates won the lottery. *(A)*
Bill Gates is a millionaire. *(B)*
If Bill Gates wins the lottery he will be a millionaire. *(If A Then B)*

Figure 2.

Before we can begin to analyse an argument of the kind given in Figure 1 we need to be able to represent the argument in a formal notation suitable for computational processing. We have to accept that in order to get from a textual representation of an argument or a dialogue account to a formal representation suitable for computational processing there will have to be a pre processing step. Tools such as Araucaria, [**Reed and Rowe, 2004]**, Archelogos **(http://archelogos.com/archelogos/)** and RSTTool **(http://www.wagsoft.com/RSTTool/)** provide a method of achieving this pre processing step. Araucaria appears the best suited to performing the required pre processing as it makes use of AML in its underlying representation. AML (Argument Markup Language) as defined by Reed and Rowe is a schema for a language that can be used to markup arguments. AML is XML based which can be easily processed by computer programs. AML however; does not, in its current form, provide enough information to allow us to easily identify fallacy. Araucaria was designed as a diagramming tool to help visualise a textual argument in order to act as a learning aid for the study and everyday use of argumentation theory in various scenarios**.** This is heavily reflected in the AML schema.



---

[1] Affirming the Consequent is an invalid form of argument the valid form of which is known as "Denying the Consequent" or Modus Tollens.

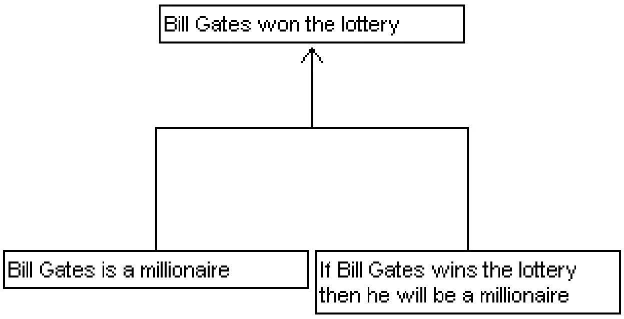


Figure 4.

Figure 4 shows the argument from Figure 1 diagrammed in Araucaria. The resulting AML is shown in Figure 5.

```
<AU>
  <PROP identifier="A" missing="no">
    <PROPTEXT offset="100">
       Bill Gates won the lottery
    </PROPTEXT>
    <ROLE class="wigmoreFact" element="none"/>
    <ROLE class="addedNegation" element="no"/>
    <ROLE class="toulmin" element="data"/>
    <ROLE class="wigmore" element="evidenceTestAffirm"/>
    <TUTOR end="0" start="0"/>
  </PROP>
  <LA>
    <AU>
      <PROP identifier="B" missing="no">
        <PROPTEXT offset="62">
           Bill Gates is a Millionaire
        </PROPTEXT>
        <ROLE class="wigmoreFact" element="none"/>
        <ROLE class="addedNegation" element="no"/>
        <ROLE class="toulmin" element="data"/>
        <ROLE class="wigmore" element="evidenceTestAffirm"/>
        <TUTOR end="0" start="0"/>
      </PROP>
    </AU>
    <AU>
      <PROP identifier="C" missing="no">
        <PROPTEXT offset="0">
           If Bill Gates wins the lottery then he will be a
           millionaire
        </PROPTEXT>
        <ROLE class="wigmoreFact" element="none"/>
        <ROLE class="addedNegation" element="no"/>
        <ROLE class="toulmin" element="data"/>
        <ROLE class="wigmore" element="evidenceTestAffirm"/>
        <TUTOR end="0" start="0"/>
      </PROP>
    </AU>
  </LA>
</AU>
```

Figure 5.

The problem with using AML for our purposes is that we need a finer level of detail to be expressed about the structure of the argument we are marking up. The information we require is similar to the information we obtain by writing the argument from Figure 1 using propositional logic. However, we want to preserve the relationship that AML captures between the original textual argument and the marked up argument. This prevents us from simply converting our argument into a PROLOG representation for example.

The goal therefore of our pre-processing step is to produce a marked up argument that gives us the formal processing benefits of propositional logic with the less formal but human readable and easily editable features of AML, while maintaining the link with the original textual argument. Given that AML has certain desirable features that we would like to include in our new method of marking up argument it seems sensible to extend AML to include the small additions that would make identification of formal fallacy possible rather than constructing a new markup language from the ground up.

The important piece of information that is lost in our Araucaria representation of the textual argument from Figure 1 is that node C in our Araucaria diagram no longer has the knowledge of being an implication operating on the other two propositions. What is meant by this is that in our textual argument the proposition that states “If A Then B” becomes node C in our Araucaria diagram. This is an important aspect in the identification of the fallacy of Affirming the Consequent but in marking our argument up in AML we have lost this information. If we can however extend AML to include a representation capturing the idea of AML nodes being logical operations on other AML nodes then we will have all the information necessary to identify the argument from Figure 1 as an instance of the fallacy of Affirming the Consequent. Figure 6 below shows the AML from Figure 5 modified to include an extension that captures the knowledge of proposition C being an instance of a propositional logic implication.

```
<PROP identifier="C" missing="no">
  <PROPTEXT offset="0">
    If Bill Gates wins the lottery then he will be a
    millionaire
  </PROPTEXT>
  <ROLE class="wigmoreFact" element="none"/>
  <ROLE class="addedNegation" element="no"/>
  <ROLE class="toulmin" element="data"/>
  <ROLE class="wigmore" element="evidenceTestAffirm"/>
  <TUTOR end="0" start="0"/>
  <PropositionalLogicOperation type="implication">
    <LeftOperand id="A"/>
    <RightOperand id="B"/>
  </PropositionalLogicOperation>
</PROP>
```

Figure 6.

AML already captures the structure of the argument and assigns IDs to the propositions so the extension shown in Figure 6 is unobtrusive. It captures the idea that proposition C has a larger role to play in this argument than could previously be identified using the original AML schema.

Araucaria can be extended to allow an analyst to be able to markup an argument as shown as easily as they markup an argument using the current AML schema. Araucaria maintains a link between the original text and the text associated with each node in the diagram. Allowing an analyst to assign a *PropositionalLogicOperation* to an AML node would be an extension to the current Araucaria user interface. By processing the AML in the diagram already Araucaria could work out given the text in the node that was assigned a *PropositionalLogicOperation* what other nodes were being operated on and automatically insert the correct *LeftOperand* and *RightOperand* elements into the AML.

With our textual argument formally represented we have the input in a computationally acceptable format. In order to identify the fallacy of Affirming the Consequent we still need to have a formal representation of the pattern that the fallacy exhibits. XML can be used to represent this pattern. Figure 7 shows an example of Affirming the Consequent expressed as XML.

```
<Fallacy name="Affirming the Consequent" type="formal">
  <PropositionalLogicOperation type="implication">
    <LeftOperand  id="A"/>
    <RightOperand id="B"/>
  </PropositionalLogicOperation>
  <LogicalError>
    <Conclusion  value="A"/>
    <Proposition value="B"/>
  </LogicalError>
</Fallacy>
```

Figure 7

With both the input and the fallacy template represented in a manner that can be easily processed by a computer program the actual identification of the instance of the fallacy of Affirming the Consequent shown in Figure 1 is a relatively straight forward matching of the argument against the fallacy pattern. This approach can be extended to identify formal fallacies such as Denying the Antecedent, Denying a Conjunct and Affirming a Disjunct, essentially fallacies that exhibit an error in their logical form.

3 Applications

An application of a computational method for identifying fallacy is in a learning environment for use by students of argumentation, philosophy, law and computing amongst others. Having the ability to analyse arguments for the occurrences of fallacy would be beneficial for students gaining an understanding of how fallacy is used in everyday arguments. Being able to tailor the matching criteria could enable students to see how the views on fallacy have varied from the work by Aristotle through to Hamblin, Mackenzie and Walton. This is what we will term batch analysis of argument in the sense it is not performed in real time. Batch analysis of arguments can also have applications in industry. Fields such as politics and law could benefit just as they benefit from the ability to diagram arguments using a tool such as Araucaria.

Real time analysis and visualisation of arguments in a legal case could act as a tool for lawyers in the courtroom. Students of argumentation could be able to see in real time how fallacies occur in arguments. Real time simulation of debates and argument can highlight how fallacy manages to go undetected in everyday use. The most interesting applications manifest themselves when we consider performing fallacy identification in real time. Dialogue and argumentation play an important role in Multi Agent Systems [**McBurney and Parsons, 2002**]. Current multi agent systems make use of communication protocols whereby they engage in dialogue to perform complex tasks. Agents engaging in dialogue have no way at the moment to determine if the arguments they are receiving are fallacious or not. An agent equipped with the ability to identify instances of fallacious arguments and make use of fallacy as an argumentation technique are better positioned to know arguments that are acceptable and have access to a wider range of argumentation strategies.

4 Extensions

The approach to fallacy identification we have discussed is; in its present form, simplistic in nature. We make use of AML which is designed to capture the argument expressed in a monologue. Given the work done by [**Hamblin, 1970**], [**Mackenzie, 1979**], [**Walton, 1994**], [**Woods and Walton, 1978**] to look at fallacies within a dialectical framework it seems sensible for us to extend our approach in a similar fashion to work within the context of a dialogue. To extend our model into a dialectical setting is something we aim to achieve in future work.

[Walton, 1994] argues that the distinction between a formal and informal fallacy is not as simple as we would hope and the boundary between formal and informal fallacy is blurred but we know that our approach is not currently designed to cater for interpretations of formal fallacy other than the traditional interpretation which is that a formal fallacy is an error in the logical form of the argument. Extending the model to deal with other interpretations of formal fallacy may be achieved by implementing a system whereby the fallacy templates can be specified and added to a database of fallacy templates that the identification system can use to match arguments against.

An area for further research would be to extend our current approach to be able to deal with informal fallacies. Informal fallacies are more complex and simply analysing arguments for invalid use of propositional logic is not enough. The fallacy template would have to be extended to allow other aspects of argument and in particular dialogue to be specified for analysis. For example being able to include the dialogue type the fallacy is likely to occur in, the pre conditions such as entries in a players commitment store(s) and sequences of moves that indicate the presence of a fallacy, the idea of dialogue shifts and other aspects are all important and may help extend our approach to cope with informal fallacy.

5 Conclusion

We have introduced a simple model for identifying formal fallacy. Using Affirming the Consequent as an example we have shown how it is possible to markup a textual argument in such a way that identification of instances of formal fallacies is possible in a computational setting. Our approach is based on the approach of using fallacy as an argumentation technique in the style of Walton rather than attempting to prohibit its use as in the work done by Hamblin and Mackenzie. Although the method discussed is simple and still in its infancy there are many possible extensions and applications. We have discussed some of the possible applications of a functioning computational system for identifying fallacy and also explored extensions to our current approach. We have currently made use of AML as the basis for the markup of our arguments, although this may not be the best approach. AML was used as the modifications to the schema were minor and a tool was currently available to produce AML for testing our methods. Araucaria could be extended easily to allow us to markup our arguments in the way we have shown. Another benefit is that Araucaria would be able to do a lot of the processing work to automatically decide which propositions became the operands of the *PropositionalLogicOperation* XML node due to the link between the AML markup and the text of the argument. It follows then that users of Araucaria would simply have to indicate a node as being a *PropositionalLogicOperation* of type "x" and Araucaria would be able to carry out the rest of the processing. Further work is however required to consider markup of arguments with respect to fallacy. In particular work by [**Rahwan and Sakeer, 2006]** could provide an alternative solution to the markup language we use.

The complicated nature of fallacy makes producing a holistic computational solution a challenging task. We have started with a small area of fallacy but by starting with a simple model for identifying fallacy and extending it slowly to cope with more complex scenarios the hope is to gain a better understanding of how to tackle fallacy computationally and produce a system that allows fallacy to be used as an argumentation technique in current computer based multi agent systems.

References

[Aristotle, 1955] Aristotle. On Sophistical Refutations. *Trans. E. S. Forster, Loch Classical Library, Cambridge, Mass., Harvard University Press,* 1955.

[Hamblin 1970] C. L. Hamblin. Fallacies. *London: Methuen,* 1970.

[Woods and Walton, 1972] John Woods and Douglas Walton. "On Fallacies". *Journal of Critical Analysis 4*:103-11, 1972.

[Mackenzie, 1979] Mackenzie, J. D. "Question Begging in Non-Cumulative Systems". *Journal of Philosophical Logic 8*:117-33. (1979)

[vanEemeren and Grootendorst, 1984] van Eemeren, Frans H., Rob Grootendorst. "Speech Acts in Argumentative Discussions". *Dordrecht: Foris Publications,* 1984

[Walton and Batten, 1984] Douglas N. Walton and Lynn M. Batten. "Games, Graphs and Circular Arguments". *Logique et Analyse 106:133-164,* 1984.

[Walton, 1994] Walton, Douglas N. "Begging the Question as a Pragmatic Fallacy". *Synthese 100*:95-131, 1994.

[Walton, 1995] Walton, Douglas N. "A Pragmatic Theory Of Fallacy". *University of Alabama Press,* 1995.

[McBurney and Parsons, 2002] McBurney P. & Parsons S. "Games that agents play: A formal framework for dialogues between autonomous agents", *Klawer Academic Publishers,* 2002.

[Reed and Rowe, 2004] Reed, C.A. & Rowe, G.W.A. "Araucaria: Software for Argument Analysis, Diagramming and Representation". *International Journal of AI Tools* 14(3-4):961-980, 2004.

[Godden and Walton, 2005] Godden, D., Walton, Douglas N. "Denying the Antecedent as a Legitimate Argumentative Strategy - A Dialectical Model". *Journal of Informal Logic Fall 24(3),* 2005.

[Rahwan and Sakeer, 2006] Rahwan, Iyad Sakeer, P. V. "Representing and Querying Arguments in RDF". *CMNA 6 Riva del Garda Italy*, 2006.

[Wells and Reed, 2006] S. Wells and C. Reed. "Knowing When to Bargain". *6th International Workshop on Computational Models of Natural Argument. COMMA, Liverpool, U.K., 2006.*