

INTRODUCING ALIAS

Simon Wells & Roberto La Greca

Edinburgh Napier University

CMNA 15 @ PRIMA 2015 (Bertinoro, Italy)



ROBERTO

Summer Research Internship

INTRODUCTION

- Increasingly necessary to work with arguments digitally
- Many questions & techniques, e.g. in Argument Mining, are predicated on access to large amounts of data (for training, testing, &c.)
- There are tools to support working with arguments (but many have drawbacks)
 - Araucaria - old, abandoned (more or less) difficult to integrate within an automated workflow/not scriptable
 - ARG-Tech toolset - fine & functional but non-free & reliant on *other peoples computers*
 - Plethora of Abstract tools (e.g. 18 free & open source tools submitted to the recent First Competition on Computational Models of Argumentation)
 - Carneades - functional, performant, free - tendency to change implementation language, requires adherence to the Carneades structured model of argument
 - Non-mainstream implementations
- Impact: Getting outputs of research into hands of (end-)users/developers/other researchers
- An eco-system of tools is healthy
- Why not one more? - a tool that aims to bridge between existing *problem specific* tools (whilst being free, open, flexible)

MOTIVATION

- Support abstract & natural argumentation
- Bridge between domains
- Provide composable tools that enable arguments to be defined, stored, analysed, manipulated, explored
 - Build a *swiss army knife* for working with argument resources, e.g.
 - create|import arguments
 - persist arguments as files or within databases (choice of DB consistent with nature of problem)
 - analyse|manipulate arguments
 - share arguments
 - Try not to specify a particular approach (argument model)
- Use within research & teaching (& ideally outwith the field)

ALIAS

- Pure Python library - Batteries included (but optional extras)
 - for working with arguments (predominantly abstract but not by design (& not forever))
- Multi-contextual use cases
 - Standalone toolset (scriptable)
 - Programming library (incorporate into other tools)
 - Interactive workbench (via iPython (Jupyter) notebooks)

WORKING WITH ARGUMENTS

- Serialisation - Aim to be an *interlingua* for arguments - can read in & translate between:
 - Aspartix “apx”
 - Trivial Graph Format “tgf”
 - Dot Language
 - JSON
 - NetworkX native graph formats
- API including support functions/algorithms including:
 - Calculations of abstract semantics & labellings for input graphs that contain “attacks” relations, e.g.
 - Complete, grounded, preferred, stable, semi-stable

PERSISTING ARGUMENTS

- SQLAlchemy provides a database abstraction layer
 - Currently out-of-box support for SQLite & Neo4j
- Choice of datastore can make a huge difference to performance so aim to provide flexibility in selecting/importing/exporting data between datastores (e.g. Postgres, CouchDB, MongoDB)

ALIAS & IPYTHON NOTEBOOKS

- An enhanced REPL for working with Python
- Runs a local-server that serves up the notebook & enables interaction via a standard, modern web browser
- Mix data & code within same “notebook” interface
 - Used increasingly within Physics & Biological Sciences labs for recording protocols, analysing & exploring data - has assumed role of digital lab notebook - even being used to present results/papers incorporating data and prose enabling **reproducibility** of research findings
 - Plain text so can be stored in a Git (or other) repository for sharing between members of group
 - Used pedagogically to provide questions/exercises/workbooks to students - calculate the answers and show your working (results are live when the notebook is run)
 - Can use any Python library (& libraries that Python can proxy),
 - e.g. we model arguments as graphs so are leveraging networkX as much as possible to do graph analysis (& visualisation)
 - Javascript to enable interactive visualisations
 - e.g. D3.js based Javascript widget to enable interactive visualisations (Argument serialised as JSON then) - can take your data out of the pipeline and incorporate into “other” web resources

Welcome to the ALIAS Demonstration Notebook!

This Ipython Notebook aims to demonstrate the key functionality of the ALIAS library.

```
In [1]: import alias as al
example = al.ArgumentationFramework('Example')
```

Lets add a few arguments to the framework we've called 'Example':

```
In [2]: example.add_argument('a')
example.add_argument('b')
example.add_argument('c')
# Arguments can also be passed as a list or tuple
# e.g: example.add_argument(['a', 'b', 'c'])
```

Now, lets create some attacks between these arguments:

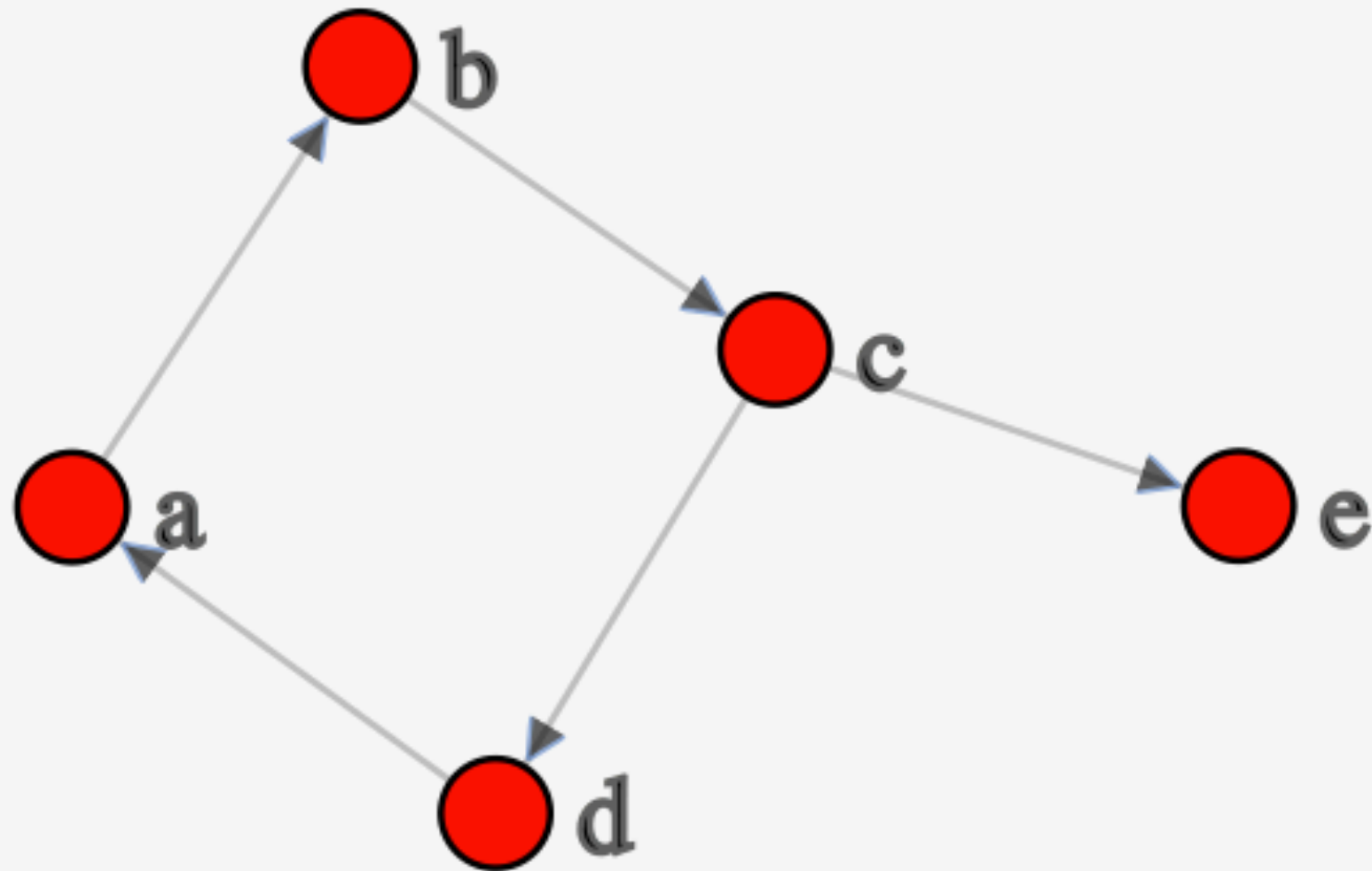
```
In [4]: example.add_attack(('a', 'b'))
example.add_attack(('b', 'c'))
# Attacks can also be passed as a list or a tuple
# by using the optional parameter 'atts'
# e.g: example.add_attack(atts=[('a', 'b'), ('b', 'c')])
```

We have created an Argumentation Framework called example which contains three arguments and two attacks. For a string representation of the Framework, simply call print on it:

```
In [5]: print example
ArgumentationFramework 'Example' : {'a' : ['b'], 'c' : [], 'b' : ['c']}
```

Argument objects belonging to a framework can be referenced by name like so:

```
In [7]: arga = example['a']
print example.get_attackers('b')
print arga
set(['a'])
Argument: 'a' : ['b']
```



JAVASCRIPT VISUALISATION WIDGET

NB. AIF based visualisation. Can also show locutional content of nodes as well as support for Info (I) nodes, rule applications (RA-nodes), conflict application (CA nodes) , & archetypal schemes (S-nodes)

FUTURE WORK

- Early but active development
- Increase feature-set - specifically natural language pipeline
 - Natural Language > Markup/Annotation > Analysis/Manipulation > Storage > Visualisation
 - Unified, stable API
- More increased support for Abstract tools
 - Choice of implementations (clarity versus performance)
 - Performance improvements: explore high performance scientific computing library (NumPy, SciPy) for implementation of algorithms
- Plans to use as a teaching aid within a computational argumentation module (evaluated by students)

CONCLUSIONS

- Yet Another Argumentation Tool
- Attempt to provide a comprehensive (ongoing) toolset for working with arguments
- Aim to support researchers, students, developers, & end-users

THANKS FOR YOUR ATTENTION

ANY QUESTIONS?