Monkeypuzzle

Towards Next Generation, Free & Open-Source, Argument Analysis Tools

John Douglas
Edinburgh Napier University
Merchiston Campus
Ediburgh EH10 5DT
john@johndouglas.co

Simon Wells
Edinburgh Napier University
Merchiston Campus
Ediburgh EH10 5DT
s.wells@napier.ac.uk

ABSTRACT

We introduce a new, free, open-source, web-based argument analysis tool called Monkeypuzzle. This is designed to provide both a foundation for creating and visualising reproducible argument analyses as well as a flexible framework for investigating new and extending existing argument analysis techniques.

CCS CONCEPTS

• **Computing methodologies** → **Discourse, dialogue and pragmatics**; *Nonmonotonic, default reasoning and belief revision*; • **Information systems** → *Web interfaces*;

KEYWORDS

Argument Analysis, Open Source Tools, Argument Visualisation

1 INTRODUCTION

Monkeypuzzle is a web-based tool, following an open development model, with a focus on pure argument analysis, support for flexible deployment, and rapid innovation with respect to both argument analysis and visualisation techniques. A range of newer features have been developed that go beyond the extant tools to address some shortcomings and to support the needs of changing analytical endeavors. The initial feature set has been spurred by ongoing work to develop the Sustainable Transport Communications Dataset (STCD[1]) [9], an effort to develop a large-scale, high quality analysis of arguments used within sustainable transport communication for behaviour change. During these efforts it became apparent that a modern, free, and open-source argument analysis tool was required that could meet the needs of contemporary argument analysts, based upon an open development and deployment model that could sustain rapid, demand-driven innovation.

2 RELATED WORK

There have been a range of argument analysis tools published over the years including Araucaria[6], Rationale[2], Ova/Ova+[3], as well as tools that have supported aspects of argument analysis within more complex workflows, for example Debategraph[4] amongst many others. See [2] for a bibliography of argument diagramming tools. Monkeypuzzle has been inspired by this rich heritage of past argument analysis tools, indeed it's name is an homage to the common name of the Araucaria tree. Monkeypuzzle adopts those elements that are both familiar and useful from existing tools, such as the two pane, source text pane and analysis canvas pane, UI pattern introduced with Araucaria [6]. The specific boxes and arrows visualisation is a variation on the *de facto* Argument Interchange Format (AIF) [1] layout found in the OVA/OVA+ tool, utilising circles to depict *I*-Nodes and diamonds to depict *S*-Nodes.

3 MONKEYPUZZLE

Monkeypuzzle is a free, open source, browser-based argument analysis tool that has the following features:

(1) Complete source-code available under a permissive license - Full source code is available from the ARG@ENU GitHub project repository[5] under the GPL3 license[6]. The importance of this is twofold. Primarily, users can build the app into their workflow without risk that it subsequently either becomes unavailable or only available under a restrictive or expensive license. Secondarily, because the source is available, users can host their own instances and enhance the app to include features that fit their own research goals; Monkeypuzzle thus becomes a platform not only for research but also for experimentation with new argument analysis and visualisation techniques.
(2) Multiple deployment options - The primary mode of interaction with the app is via the hosted deployment[7] however the app is not server dependent and two offline forms are supported. The app can be run from a local filesystem by loading the index.html file into a browser. An offline version is also supported so that the app is cached in the users browser and reloads from there when the user navigates to the app's URL, even if the user is offline.
(3) Simultaneous analysis of multiple source texts - This is the main innovation within the Monkeypuzzle user interface. Multiple source texts, currently set

[1] https://github.com/ADAPT-project/STCD
[2] http://www.reasoninglab.com/
[3] http://ova.arg-tech.org/
[4] http://debategraph.org
[5] https://github.com/ARG-ENU/monkeypuzzle_web
[6] https://www.gnu.org/licenses/gpl-3.0.en.html
[7] http://arg.napier.ac.uk/monkeypuzzle/

to an arbitrary maximum of ten, can be loaded into individual tabs on the text panel and a single analysis made within the visualisation panel. This enables a domain analysis to be created from multiple resources something that is difficult to do with other tools.

(4) Support for canonical representations of text nodes - When analysing multiple source texts and attempting to create a single, large domain analysis rather than a series of individual analyses, variations in voice, writing styles, complexity of language, and completeness of utterance can reduce the coherency and fluidity of the resulting dataset. The app supports editing of node text into a canonical form whilst also saving the original expressions. This enables higher quality, curated argument datasets to be constructed. This is particularly important as argument research foci move from straightforward argument analyses towards reuse of the resultant datasets, for example, in natural language generation tools or to support exploration of contentious knowledge domains.

(5) Serialisation to a simple JSON format - The needs of the tool are driving development of a simple, native, JSON-based file format for saving and loading analyses. The aim is to identify new, useful criteria that can be used to support extension and improvement of the AIF. Whilst support for the AIF is on the project's roadmap, it was decided that a more appropriate starting point would be to rapidly account for the various kinds of metadata that the STCD analysis work is uncovering. User research during our development has shown that many researchers who are performing argument analysis desire the ability to make *ad hoc* collections of metadata, as demanded by their data, and suggest that current tools frustrate this desire.

(6) Export to graphics formats - Visualisations can be saved for reuse in other contexts using the Portable Network Graphics (PNG) and Scalable Vector Graphics (SVG) formats.

(7) Support for hierarchically organised Argumentation Schemes - Walton and Macagno propose a hierarchical organisation of Argumentation Schemes [8] which is implemented within the app. This gives structure to the user and aids in the selection of a scheme to assign to an argument, rather than choosing from a long list, organised only by scheme set, a user is able to select a scheme from a range of categories to drill down to an appropriate scheme. The goal is to make it easier to select a scheme to characterise an argument by so that more argument analyses contain comprehensive scheme analyses rather than extensive use of the "default" scheme.

Bootstrapping a new argument analysis tool to this point has taken significant effort. Much of the existing work has been preliminary scaffolding to enable the future implementation, integration, exploration, and maintenance of both new and refined analysis procedures. The authors do not intend to suggest that the current application is particularly innovative; beyond the bringing together of a core selection of proven argument analysis techniques in anticipation of a growing community of developers who might take the app in directions contrary to those mapped out in the remainder of this paper.

4 CONCLUSIONS & FUTURE WORK

The full roadmap is detailed online[8] and the project is under active development. Immediate development goals are as follows: to exploit the use of a tested, reliable, and scalable Javascript graph layout library, such as d3.js[9] or cytoscape.js[10], so that argument graphs can be automatically rendered to the screen, minimising the need for users to manually adjust the placement of nodes. Additionally we aim to support mapping of selections from disparate source texts onto the same analysis nodes, effectively merging nodes that have the same meaning but different natural language expressions, especially where these have originated from different resources. The aim here is to support the development of large, high quality, and integrated argument maps and corpora across domains rather than being restricted only to the analysis of a single given source at a time. The resource pane, although currently restricted to textual resources, will eventually support analysis of arguments from a variety of file types, for example, parsing web-pages (HTML), Portable Document Format (PDF), video, and audio files, to enable multi-modal argument analysis.

Three areas of active research that we are pursuing are, firstly, the integration of modified versions of storymaps that incorporate argument structure, secondly, support for effective dialogue analysis, and thridly, support for visualisation at scale. Storymaps are Geographic Information Systems (GIS) that integrate cartographic maps, geospatial data, and narrative driven content. In 2012, ESRI, a developer of GIS and spatial analytics software, introduced storymaps and went on to win awards for Best Digital Map Product and Best Overall Map Product from the International Map Industry Association. Storymaps have since been used to good effect in many journalistic contexts and many nice examples can be viewed at the Storymaps website[11] however an area that has not been exploited is the combination of argumentative data and metadata with specific locations and journeys so that arguments can be visualised in the context of the geographic locations that they relate to. We believe that this could prove to be a useful new dimension in the context of how legal argument, particularly witness testimony, is explored and visualised. Dialogue analysis has not been well supported by the open-source argument analysis tools but the links between argument and dialogue have been recognised for many years, having been explored by O'Keefe [4] in terms of Argument_1 and Argument_2, or argument as process and argument as product, but also more recently in dialogical extensions to

[8] https://github.com/ARG-ENU/monkeypuzzle_web/issues
[9] https://d3js.org/
[10] http://js.cytoscape.org/
[11] https://storymaps.arcgis.com/en/

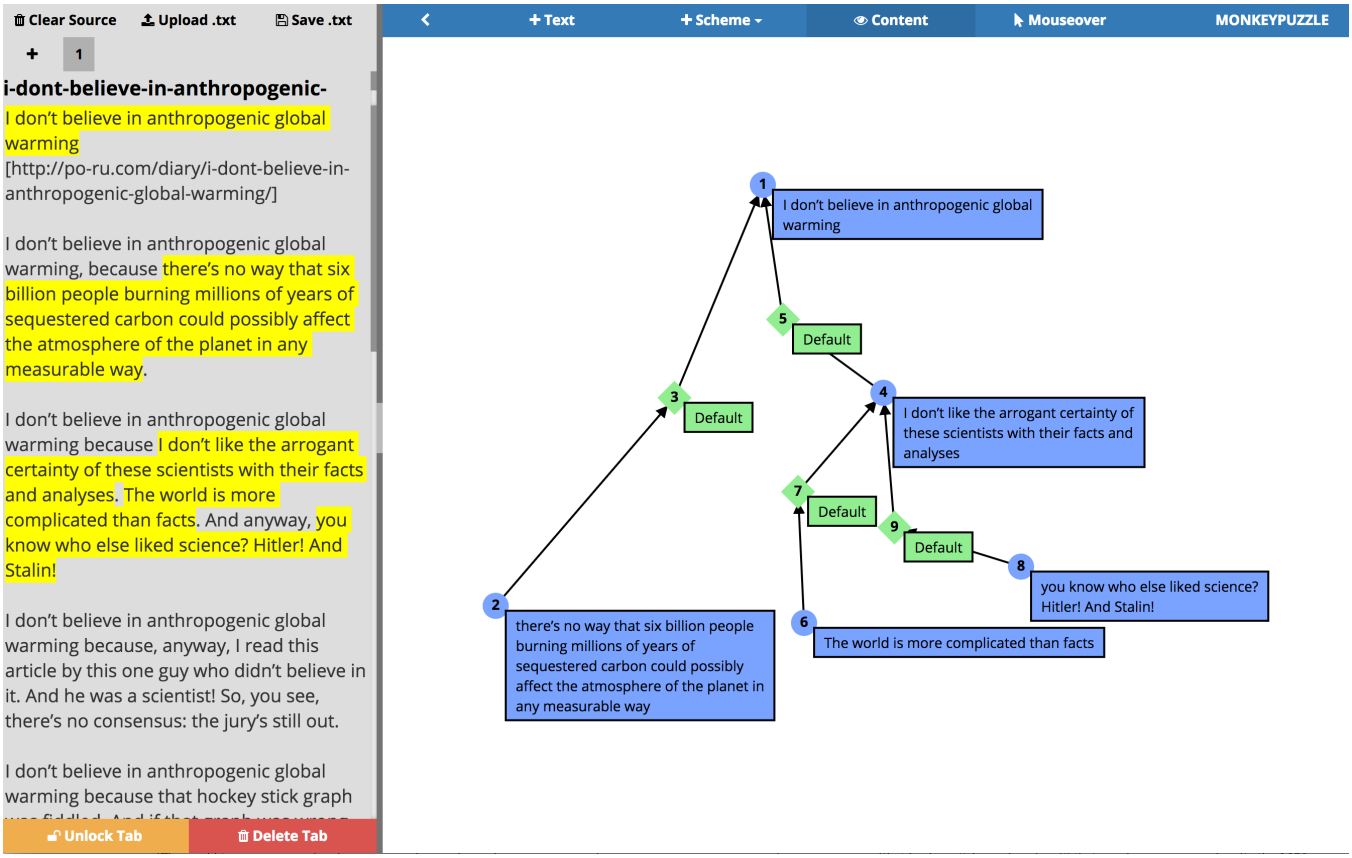

Figure 1: The default Monkeypuzzle User Interface showing the standard, two-pane UI popularised by Araucaria. The left-hand pane is the source pane, a tabbed collection of textual resources for analysis. The right-hand pane is the visualisation pane. The source pane can be completely collapsed to give a user more room to freely create an argument diagram independent of any specific source text allowing the app to be used for argument construction and exploration as well as argument analysis.

the AIF [7] which operationalises the co-construction of argument as a product of dialogue. One approach might be to enable dialogues to be annotated according to the rules of established dialectical games [10] and for the argumentative content licensed by the moves within the dialogue, for example statement→challenge→defense sequences, to be extracted into the visualisation. Finally, visualisation at scale will increasingly become an issue as the sizes of argumentative datasets and corpora increase. Anecdotally, standard box and arrow diagrams often become unwieldy to the point of unusability at around the 50 to 100 node mark. Yet the combined output from increasingly accurate Argument Mining tools [3], or the fulfilled promise of the Argument Web [5] will yield argument datasets at a scale where the limits of current visualisation tools are exceeded.

Ultimately we plan for Monkeypuzzle to provide a basis for exploring new argument visualisation techniques, to act as a test-bed for new tools to interact with argumentative datasets, and to contribute to a healthy and varied ecosystem of argument tools to support further development of computational models of argument.

REFERENCES

[1] C. Chesnevar, J. McGinnis, S. Modgil, I. Rahwan, C. Reed, G. Simari, M. South, G. Vreeswijk, and S. Willmott. 2006. Towards an Argument Interchange Format. *Knowledge Engineering Review* 21, 4 (2006), 293–316.

[2] D. Khartabil, S. Wells, and J. Kennedy. 2016. Large Scale Argument Visualization (LSAV). In *Proceedings of EUROGRAPHICS 2016*.

[3] M. Lippi and P. Torroni. 2016. Argumentation Mining: State of the Art and Emerging Trends. *ACM Transactions on Internet Technology (TOIT)* 16, 2 (2016).

[4] D. J. O'Keefe. 1977. Two Concepts of Argument. *The Journal of the American Forensic Association* 13, 3 (1977), 121–128.

[5] I. Rahwan, F. Zablith, and C. Reed. 2007. Laying the Foundations for a World Wide Argument Web. *Artificial Intelligence* 171 (2007), 897–921.

[6] C. Reed and G. Rowe. 2001. *Araucaria: Software For Puzzles In Argument Diagramming And XML*. Technical Report. University Of Dundee.

[7] C. Reed, S. Wells, G. W. A. Rowe, and J. Devereux. 2008. AIF+: Dialogue in the Argument Interchange Format. In *Proceedings of the 2nd International Conference on Computational Models of*

Argument (COMMA 2008).

[8] D. Walton and F. Macagno. 2016. A classification system for argumentation schemes. *Argument and Computation* 6, 3 (2016), 219–245.

[9] S. Wells and K. Pangbourne. 2016. Using Argumentation Within Sustainable Transport Communication. In *Argumentation and Reasoned Action, Proceedings of the 1st European Conference on Argumentation, Lisbon 2015, volume 1*. College Publications, Chapter 34, 781–801.

[10] S. Wells and C. Reed. 2012. A domain specific language for describing diverse systems of dialogue. *Journal of Applied Logic* 10, 4 (2012), 309–329.