1. Introduction

Contrary to unstructured representations like natural language texts, argument graphs enable advanced analysis of an argument’s structure which consists of linked Argumentative Discourse Units (ADUs). Since most existing works dealing with the creation of such graphs are primarily geared towards experts and neglect the needs of developers and laymen, we propose (i) an intuitive, stable, and scalable tool (ArgueMapper) for creating and browsing graph-based representations of arguments by experts and laymen alike and (ii) a straightforward format (Arguebuf) enabling developers to build related tools and exchange data more easily. Both ArgueMapper and Arguebuf are available under the permissive MIT license and are open to any kind of contribution.

2. ArgueMapper: A Tool for Manual Argument Mining

This section will highlight some features of ArgueMapper compared to existing tools like Online Visualization of Arguments (OVA) [1] and MonkeyPuzzle [2].

Intuitive Interface  Our tool (see Figure 1) complies with Nielsen’s usability heuristics [3] to ensure as little friction as possible for laymen. At the same time, it is similar enough to OVA to be familiar to experts as well.

Optimized for Mobile Devices  ArgueMapper is fully functional on smartphones and tablets by providing finger-optimized buttons and gesture controls.

Auto-Layout  We combined ideas of OVA and MonkeyPuzzle by implementing a hierarchical automatic layout algorithm that runs entirely in the user’s browser.

State Management  To prevent loss of unsaved data, the app’s state is always stored in the browser’s storage. In addition, we also fully support undo/redo functionality.

Modern Development Stack  To simplify contributions, we built ArgueMapper using modern tooling like TypeScript and React. It has a modular architecture and thus may be embedded into other systems as well.
3. Arguebuf: A Format for Argument Graphs

In conjunction with ArgueMapper, we developed the format Arguebuf to address limitations of existing ones like Argument Interchange Format (AIF) [4] and SADFace [5].

Simple Specification Arguebuf is specified using the concise and intuitive language Protocol Buffers (Protobuf), meaning that it is easily expandable.

Superset of AIF and SADFace It is possible to transform every AIF graph or SADFace document into our new format without any information loss.

Code Generation Protobuf automatically creates native code for most programming languages. Among others, this enables code completion and type checks in IDEs.

Straightforward Integration into APIs One can use a JSON-based representation for REST-APIs or utilize its binary format with gRPC to benefit from strict types.

Supercharged Python Implementation We provide an optimized Python client with advanced analysis features—for instance, importing legacy formats, converting from/to AIF, and integrating with Graphviz, NetworkX, and spaCy.

Acknowledgements

This work has been funded the by the DFG within the project ReCAP-II (No. 375342983) as part of the priority program RATIO (SPP-1999) as well as the Studienstiftung.

References